



**Rui Miguel Arantes
Fernandes Palha**

**Interacção entre um dispositivo móvel e um ecrã de
grandes dimensões**



**Rui Miguel Arantes
Fernandes Palha**

**Interacção entre um dispositivo móvel e um ecrã de
grandes dimensões**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações (M.I.E.E.T.), realizada sob a orientação científica do Professor Doutor Paulo Miguel de Jesus Dias, Professor Auxiliar e da Professora Doutora Maria Beatriz Alves de Sousa Santos, Professora Associada com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

O Júri

Presidente

Doutor João Nuno Pimentel da Silva Matos

Professor Associado do Departamento de Electrónica e Telecomunicações e Informática da Universidade de Aveiro

Orientador

Doutor Paulo Miguel de Jesus Dias

Professor Auxiliar do Departamento de Electrónica e Telecomunicações e Informática da Universidade de Aveiro

Co-Orientadora

Doutora Maria Beatriz Alves de Sousa Santos

Professora Associada com Agregação do Departamento de Electrónica e Telecomunicações e Informática da Universidade de Aveiro

Arguente

Doutor José Francisco Creissac Freitas Campos

Professor Auxiliar do Departamento de Informática da Escola de Engenharia da Universidade do Minho

Agradecimentos

Dedico este trabalho à minha mãe, pelo esforço que fez para me proporcionar esta oportunidade e pelo seu apoio incondicional.

Ao meu irmão que sempre me deu a alegria de um fim-de-semana.

Ao meu pai que me acompanhou durante o meu percurso académico.

A toda a minha família que sempre me proporcionou bons momentos de convívio.

Ao Falcão, ao Alex, ao Adriano, ao Diogo Cunha, ao Diogo Dias, ao William, ao Yann, ao Quintas, ao Rómulo, ao Paulo e ao Aníbal pela amizade e companheirismo ao longo da vida académica.

Aos meus orientadores, Prof. Paulo Dias e Prof. Beatriz Sousa Santos, pela disponibilidade e dedicação ao longo deste projecto.

Aos colaboradores e administração da *Ubiwhere*, em especial ao Eng. Rui. A. Costa, ao Eng. Bruno Silva e ao Eng. Ricardo Machado, pelas oportunidades que me deram de contactar com projectos e tecnologias inovadoras, assim como pelos conhecimentos transmitidos.

Palavras-chave

Computação Móvel, Ecrãs, Interação Humano-Computador, *Android*.

Resumo

A interação com sistemas multimédia interiores ou exteriores baseados em ecrãs de grandes dimensões ainda tem limitações e problemas a ser superados. Os mais recentes dispositivos móveis, equipados com sensores (tais como acelerómetros, bússolas e ecrãs sensíveis ao toque), podem ser uma alternativa para interagir com estes ecrãs já que esta solução não implica *hardware* adicional (o utilizador pode usar o seu próprio telemóvel).

Nesta dissertação apresentamos as diferentes opções e os passos que levaram à criação do DetiGuide: um guia informático para o Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro que permite aos utilizadores interagir com a informação apresentada num ecrã instalado no átrio do departamento através de um dispositivo móvel *Android*. O principal objectivo deste protótipo é estudar o uso de dispositivos móveis para interagir com ecrãs de grandes dimensões.

Keywords

Mobile Computing, Large Displays, Human-Computer Interaction, Android.

Abstract

Interactivity with outdoor and indoor multimedia systems based in large displays still has limitations and problems to overcome. More recent mobile phones, equipped with sensors (such as accelerometers, compasses and touch screens), may be an alternative to interact with those displays since this solution does not imply additional hardware (the user can use his own device). In this thesis we present the different options and the steps that led to the creation of DetiGuide: an informatics guide of the Department of Electronics, Telecommunications and Informatics of University of Aveiro that lets users interact with information displayed on a screen installed in the atrium of the department through an Android mobile device. The main objective of this prototype is to study the use of mobile devices to interact with large displays.

Índice

1.	Introdução	9
1.1.	Contexto.....	9
1.2.	Objectivos	10
1.3.	Estrutura da Tese.....	11
2.	Interacções entre Dispositivos Móveis e Ecrãs de Grandes Dimensões	13
3.	Ferramentas Utilizadas.....	23
3.1.	<i>Android</i>	23
3.1.1.	Descrição Geral.....	24
3.1.2.	<i>Software Android</i>	31
3.1.3.	<i>Hardware Android</i>	33
3.2.	<i>Eclipse</i>	37
3.3.	<i>Visual Studio</i>	37
3.4.	<i>VTK</i>	37
4.	Aplicações Desenvolvidas	39
4.1.	Arquitectura de Comunicação	39
4.2.	Testes Iniciais.....	41
4.2.1.	Controlo de um Sólido no Ecrã.....	42
4.2.2.	Jogo	43
4.3.	<i>DetiGuide</i>	44
5.	Testes com Utilizadores	49
5.1.	Usabilidade	49
5.2.	Avaliação de Interfaces.....	51
5.3.	Experiências Realizadas	52
5.4.	Resultados Obtidos	55
6.	Conclusões e Trabalho Futuro.....	63
6.1.	Análise Global	63
6.2.	Trabalho Futuro	66
	Anexo A – Código implementado em Java para um cliente – servidor TCP/IP	69
	Anexo B – Código implementado em Java para um cliente – servidor UDP.....	71

Anexo C - Instalação VTK em Windows com suporte Java	73
Anexo D – Código implementado para aceder aos valores dos sensores do telemóvel <i>Android</i>	79
Anexo E – Código implementado em Java para correcção do movimento da bola	85
Anexo F – Implementação do DetiGuide em C#	89
Anexo G – <i>DetiGuide</i> : Lista de Tarefas	97
Anexo H - <i>DetiGuide</i> : Questionário do Utilizador.....	99
Anexo I – <i>DetiGuide</i> : Dados Obtidos na 1ª Sessão de Testes com Utilizadores.....	103
Anexo J – <i>DetiGuide</i> : Dados Obtidos na 2ª Sessão de Testes com Utilizadores	115
7. Bibliografia	127

Lista de Figuras

Figura 1 - Comparação entre diferentes tipos de Interação Humano – Computador [8]...	14
Figura 2 – <i>NaviCam</i> [9].....	15
Figura 3 - Imagem obtida através do <i>NaviCam</i> [9].	15
Figura 4 - Fluxo de eventos do MAGIC Broker [10].....	16
Figura 5 - <i>Touch & Interact</i> : o utilizador toca o ecrã com o telemóvel junto da foto que pretende seleccionar, e a foto é transferida do ecrã para o dispositivo móvel [11].....	17
Figura 6 - Sistema de teste para detectar sinais do <i>C-Blink</i> [12].....	17
Figura 7 – <i>Flashlight</i> : Uma câmara debaixo do ecrã segue os movimentos do flash [13]..	18
Figura 8 - Utilizador a interagir com um dos <i>SmartBoards</i> [14].....	19
Figura 9 - Arquitectura de comunicação implementada no <i>TiltRacer</i> [15].....	20
Figura 10 - <i>TiltRacer</i> jogado por utilizadores [15].....	20
Figura 11 - Arquitectura de comunicação do <i>Doodle Space</i> [16].	21
Figura 12 - <i>Doodle Space</i> testado por utilizadores [16].....	21
Figura 13 - Movimento de Flexão/Extensão (figura A), Pronação/Supinação (figura B) e Desvio Ulnar/ Radial (figura C) [17].....	22
Figura 14 - Representação esquemática da arquitectura da plataforma <i>Android</i> [19].....	25
Figura 15 - Estrutura das interfaces gráficas nas aplicações <i>Android</i> [22].....	29
Figura 16 - Interface gráfica do emulador do <i>SDK Android</i> [25].....	32
Figura 17 - Interface gráfica do <i>SensorSimulator</i> [26].....	33
Figura 18 - <i>HTC Dream</i> [27].....	34
Figura 19 - Orientação dos eixos do acelerómetro do dispositivo móvel [28].....	35
Figura 20 - Orientação dos eixos da bússola digital do dispositivo móvel [28].....	36
Figura 21 - Esquema da comunicação implementada ao longo do projecto [3].....	39
Figura 22 - Situação normal: os três eixos são independentes [36].....	42
Figura 23 - <i>Gimbal Lock</i> : dois dos três eixos estão no mesmo plano, perdendo-se um grau de liberdade [36].....	43
Figura 24 - Sincronização entre o dispositivo móvel e o sólido.....	43
Figura 25 – Aspecto do jogo desenvolvido.	44

Figura 26 - Imagens da aplicação <i>Android</i> que controla o <i>Detiguide</i> : menu inicial (figura A), menu principal (figura B) e menu secundário (figura C).....	46
Figura 27 - Imagens da aplicação <i>Android</i> que controla o <i>DetiGuide</i> : menu de escolha de interacções (figura A) e menu de navegação (figura B).....	46
Figura 28 - Exemplo de um utilizador a utilizar o <i>DetiGuide</i> em modo de navegação usando o acelerómetro e a bússola digital. Os botões virtuais são visíveis no ecrã do telemóvel.	47
Figura 29 - Exemplo de um utilizador a utilizar o <i>DetiGuide</i> em modo de selecção usando o ecrã sensível ao toque.....	47
Figura 30 - Utilizadores usando o <i>DetiGuide</i>	54
Figura 31 - Utilizadores usando o <i>DetiGuide</i>	54
Figura 32 - Tempo médio total das tarefas nos testes com utilizadores utilizando a interacção através do acelerómetro/bússola digital.	56
Figura 33 - Tempo médio total das tarefas nos testes com utilizadores utilizando a interacção através do ecrã sensível ao toque.	56
Figura 34 - Grau de dificuldade das tarefas nos testes com utilizadores utilizando a interacção através do acelerómetro/bússola digital.	57
Figura 35 - Grau de dificuldade das tarefas nos testes com utilizadores utilizando a interacção através do ecrã sensível ao toque.	57
Figura 36 - Número de Erros nas Tarefas dos Testes com Utilizadores.	58
Figura 37 - Resultados dos Questionários dos Utilizadores após a realização das tarefas.	59
Figura 38 - Grau de Satisfação dos Utilizadores.....	60
Figura 39 - Preferência dos Utilizadores.....	60
Figura 40 - Janela do CMake.	75
Figura 41 – Janela do Visual Studio 2008.....	76

Lista de Tabelas

Tabela 1 - Diferenças entre o protocolo TCP/IP e o protocolo UDP [34].....	41
Tabela 2 - Dados gerais sobre os utilizadores (1ª Sessão de Testes).....	104
Tabela 3 - Resultados da Tarefa 1 (1ª Sessão de Testes).....	105
Tabela 4 - Resultados da Tarefa 2 (1ª Sessão de Testes).....	106
Tabela 5 - Resultados da Tarefa 3 (1ª Sessão de Testes).....	107
Tabela 6 - Resultados da Tarefa 4 (1ª Sessão de Testes).....	108
Tabela 7 - Resultados do Questionário do Utilizador - parte 1 (1ª Sessão de Testes).....	109
Tabela 8 - Resultados do Questionário do Utilizador - parte 2 (1ª Sessão de Testes).....	110
Tabela 9 - Resultados do Questionário do Utilizador - parte 3 (1ª Sessão de Testes).....	111
Tabela 10 - Resultados do Questionário do Utilizador - parte 4 (1ª Sessão de Testes).....	112
Tabela 11 - Resultados do Questionário do Utilizador - parte 5 (1ª Sessão de Testes).....	113
Tabela 12 - Dados gerais sobre os utilizadores (2ª Sessão de Testes).....	116
Tabela 13 - Resultados da Tarefa 1 (2ª Sessão de Testes).....	117
Tabela 14 - Resultados da Tarefa 2 (2ª Sessão de Testes).....	118
Tabela 15 - Resultados da Tarefa 3 (2ª Sessão de Testes).....	119
Tabela 16 - Resultados da Tarefa 4 (2ª Sessão de Testes).....	120
Tabela 17 - Resultados do Questionário do Utilizador - parte 1 (2ª Sessão de Testes).....	121
Tabela 18 - Resultados do Questionário do Utilizador - parte 2 (2ª Sessão de Testes).....	122
Tabela 19 - Resultados do Questionário do Utilizador - parte 3 (2ª Sessão de Testes).....	123
Tabela 20 - Resultados do Questionário do Utilizador - parte 4 (2ª Sessão de Testes).....	124
Tabela 21 - Resultados do Questionário do Utilizador - parte 5 (2ª Sessão de Testes).....	125

Lista de Acrónimos

API	- Application Programming Interface
CPU	- Central Processing Unit
EDGE	- Enhanced Data rates for GSM Evolution
GPS	- Global Positioning System
GSM	- Global System for Mobile Communications
IDE	- Integrated Development Environment
HTC	- High Tech Computer
MVC	- Model View Controller
PDA	- Personal Digital Assistant
SD	- Secure Digital
SDK	- Software Development Kit
TCP	- Transmission Control Protocol
UDP	- User Datagram Protocol
USB	- Universal Serial Bus
VTK	- Visualization ToolKit
WLAN	- Wireless Local Area Network
XML	- eXtensible Markup Language

1. Introdução

Este primeiro capítulo visa introduzir e contextualizar a interacção em sistemas baseados em ecrãs de grandes dimensões, assim como dar a conhecer os objectivos do trabalho desenvolvido. A última secção deste capítulo descreve a estrutura da tese.

1.1. Contexto

A tecnologia evolui diariamente, existindo actualmente numerosas soluções alternativas aos ecrãs dos computadores de secretária que até há pouco tempo eram quase a única solução disponível para visualização da informação. Em particular, os utilizadores podem agora criar ecrãs de grandes dimensões ligando vários ecrãs aos seus computadores, usando um projector, ou formando paredes digitais interactivas com múltiplos projectores.

No entanto, estas novas capacidades têm gerado novos desafios na investigação no que diz respeito ao design de interfaces de utilizador. A introdução de interactividade em sistemas multimédia presentes em ambientes exteriores, ao longo de uma cidade, ou em ambientes interiores, como numa exposição/museu ou numa escola, tem-se feito através da contínua adaptação de dispositivos de entrada, já que os mais utilizados, como ratos e teclados, não são a melhor solução para utilizar em áreas de passagem ou numa utilização de pé.

Uma série de projectos têm investigado o uso de dispositivos móveis para interagir

com o mundo real [1-7]. Esses projectos utilizam diversas técnicas de interacção tais como tocar ou apontar, e também métodos de comunicação diferentes (infravermelhos, marcadores visuais, *Bluetooth*, *Wi-Fi*, etc.). Entretanto, os mais recentes dispositivos móveis integram novas tecnologias (tais como acelerómetros, bússolas digitais e ecrãs sensíveis ao toque) que potenciam possibilidades de interacção adicionais. Contudo, os dispositivos móveis não são os únicos dispositivos que possuem capacidades de interacção. Por exemplo, no campo dos videojogos têm sido introduzidas novas formas de interacção, sendo o *Wii*, comando da *Nintendo Wii* que usa infra-vermelhos e um acelerómetro, um dos exemplos mais conhecidos.

No âmbito do plano curricular do 5º ano do curso de Mestrado Integrado em Engenharia Electrónica e Telecomunicações da Universidade de Aveiro, foi proposta em conjunto com a *Ubiwhere*, empresa sediada na cidade de Aveiro, que desenvolve aplicações para fins turísticos, a realização do presente trabalho de dissertação: “Interacção entre um dispositivo móvel e um ecrã de grandes dimensões”. A ideia base do projecto foi realizar interacções entre dispositivos móveis e ecrãs de grandes dimensões que estivessem colocados em diversos locais, tais como museus e espaços públicos, de forma a permitir aos utilizadores obterem informações sobre os locais onde se encontram, assim como de outros locais de interesse.

1.2. Objectivos

O trabalho desenvolvido ao longo desta dissertação centrou-se na concepção, implementação e teste de uma solução baseada num telemóvel *Android* utilizando a informação proveniente dos sensores que estes dispositivos móveis integram (acelerómetros, bússola digital e ecrãs sensíveis ao toque) para interagir com ecrãs de grandes dimensões. Fez-se ainda um estudo comparativo da usabilidade das alternativas de interacção desenvolvidas.

1.3. Estrutura da Tese

Além deste capítulo introdutório, esta dissertação é composta por mais cinco seguintes capítulos.

No capítulo 2, Interações entre Dispositivos Móveis e Ecrãs de Grandes Dimensões, descrevem-se alguns dos trabalhos existentes relacionados com a interacção de dispositivos móveis com o mundo real em geral e em particular com ecrãs de grandes dimensões.

No capítulo 3, Ferramentas Utilizadas, introduzem-se as ferramentas utilizadas ao longo do trabalho efectuado. Justifica-se ainda a escolha do sistema operativo móvel *Android*, e aprofundam-se as suas características e potencialidades.

No capítulo 4, Aplicações Desenvolvidas, descreve-se a arquitectura de comunicação implementada, as alternativas de interacção propostas, bem como as várias aplicações desenvolvidas.

No capítulo 5, Testes com Utilizadores, introduz-se o conceito de usabilidade e os métodos de avaliação de interfaces com o utilizador. Descrevem-se as experiências realizadas com utilizadores para testar as diversas formas de interacção, e apresentam-se os resultados obtidos.

Finalmente, no Capítulo 6, Conclusões e Trabalho Futuro, é feita uma análise global dos resultados obtidos e dos objectivos alcançados. Apontam-se algumas possibilidades para eventual trabalho futuro.

2. Interacções entre Dispositivos Móveis e Ecrãs de Grandes Dimensões

A Interação Humano - Computador é a área científica que estuda a interação entre os utilizadores e os computadores. Esta ocorre na interface, que inclui tanto hardware como software, como por exemplo caracteres ou objectos mostrados por *software* através do monitor de um computador, cuja entrada foi recebida por periféricos de *hardware* tais como teclados e ratos. O principal objectivo da Interação Humano - Computador é melhorar as interações entre utilizadores e computadores aumentando a sua usabilidade e tornando-os mais adequados às necessidades dos utilizadores [8].

Na figura 1 é possível observar diferentes formas de Interações Humano – Computador. A figura 1-A mostra que o ser humano interage directamente com o mundo real quando não utiliza o computador. Quando usa o computador ou o telemóvel diariamente, também não existe interação entre o computador e o mundo real, como se pode verificar na figura 1-B. As interações entre o mundo real e dispositivos móveis nas quais os dispositivos móveis podem ser usados para interagir com o mundo real ou outros computadores, como por exemplo ecrãs de grandes dimensões, são mostradas na figura 1-C, sendo esta forma de interações denominadas de realidade aumentada. Esta técnica de interação faz parte da área da computação ubíqua que trata das interações entre o

utilizador, o seu computador (dispositivo móvel) e objectos localizados no mundo real, tipicamente outros computadores (figura 1-E). O objectivo da realidade virtual (figura 1-D) é recriar ao máximo, e em tempo real, a sensação de realidade para o utilizador. A diferença entre realidade aumentada (figura 1-C) e realidade virtual (figura 1-D) é que na realidade virtual o dispositivo móvel ao interagir com o mundo real raramente é usado para aumentar o ambiente ou para dar ao utilizador a impressão de que este está numa realidade aumentada. Na figura 1-F, é possível observar que existem diversas formas de interacções entre o ser humano e o mundo real, seja directamente ou através de um ou vários computadores.

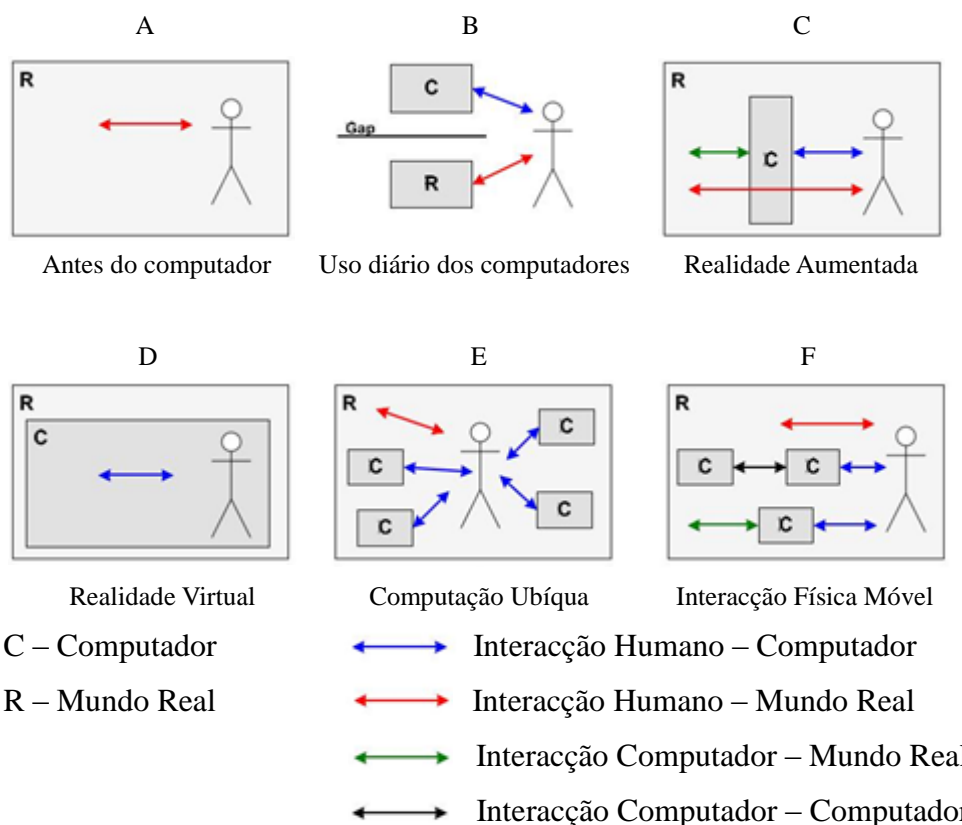


Figura 1 - Comparação entre diferentes tipos de Interacção Humano – Computador [8].

Existem diferentes definições para as diferentes formas de Interacção Humano - Computador descritos acima, o que faz com que não seja possível afirmar que uma técnica de interacção representa apenas uma forma de Interacção Humano - Computador. De seguida, são apresentados alguns trabalhos com interesse para o trabalho desenvolvido no âmbito desta dissertação.

O *NaviCam* [9] é um exemplo de um dos primeiros protótipos que encaixa na área da realidade aumentada, computação ubíqua e interacção entre o mundo real e dispositivos móveis. Este computador portátil possui uma pequena câmara de vídeo para detectar situações do mundo real. Este sistema permite ao utilizador ver o mundo real, juntamente com informações sensíveis ao contexto gerado pelo computador. Nas figuras seguintes podem-se observar o protótipo do *NaviCam* (figura 2), assim como uma imagem obtida através do dispositivo (figura 3).



Figura 2 – *NaviCam* [9].

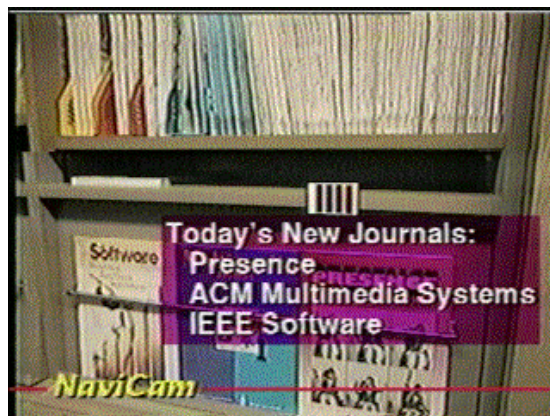


Figura 3 - Imagem obtida através do *NaviCam* [9].

Mais no âmbito desta dissertação, focada nas interacções entre dispositivos móveis e ecrãs de grandes dimensões, foram desenvolvidos vários protótipos que proporcionam diversas formas de interacção. O *MAGIC Broker* [10] é um protótipo para uma ferramenta de interacção com o público, usado em ecrãs de grandes dimensões. Está a ser desenvolvido pela *University of British Columbia*, situada em *Vancouver*, no Canadá, pela *Lancaster University*, em *Lancaster*, no Reino Unido, e pela *King Saud University*, em *Riyadh*, na Arábia Saudita. Funciona como um *middleware* entre o servidor que está ligado ao ecrã e os diferentes serviços disponíveis a partir dos dispositivos móveis. Os utilizadores podem interagir com o ecrã por voz através de um gateway *Voice XML*, enviando mensagens SMS, ou usando um browser através de um dispositivo móvel. Na figura 4 podemos ver a arquitectura do sistema onde está integrado o *MAGIC Broker* e o fluxo de eventos.

O protótipo foi integrado em algumas aplicações, entre as quais o *MashUp*, que permite 3 modos de interacção: conteúdo não-interactivo e quadro de mensagens, para

mostrar publicidade ou outro tipo de informação pública, assim como mensagens que podem ser colocadas no ecrã; pesquisa de imagens, onde o ecrã mostra as imagens pedidas pelos utilizadores que enviaram um pedido específico através do dispositivo móvel; mapas e direcções, onde um mapa do campus e direcções são mostradas no ecrã usando o *Google Maps*, e quando um browser é usado no telemóvel, também o mapa é mostrado no telemóvel.

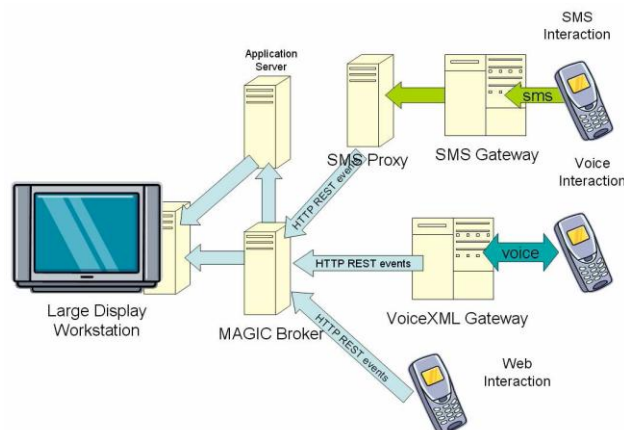


Figura 4 - Fluxo de eventos do MAGIC Broker [10].

A interacção ***Touch & Interact*** [11] permite realizar selecções num ecrã dinâmico, tocando em qualquer posição do ecrã com um dispositivo móvel.

Esta interacção tem sido desenvolvida pela Universidade de *Lancaster*, no Reino Unido e pelos laboratórios europeus da *NTT DoCoMo*, em Munique, na Alemanha. Foi desenvolvido um protótipo de um guia turístico que suporta as seguintes interacções:

- **Pairar**: usando esta interacção, o telefone pode-se mover no alcance de leitura de um item e informação adicional é mostrada no ecrã do telefone;
- **Seleccionar**: quando um item é pairado, o utilizador pode pressionar uma tecla específica no telefone para seleccionar esse item;
- **Multi-Seleccionar**: se o utilizador segurar a tecla, pode seleccionar vários itens;
- **Seleccionar por Polígonos**: pontos de polígonos podem ser desenhados segurando uma tecla específica e seleccionando o item apropriado. Quando a tecla é solta, os itens dentro da área do polígono são seleccionados;
- **Pegar e Largar**: os *itens* seleccionados são capturados usando o telefone e podem ser largados em qualquer lugar do ecrã;

- **Limpar Remoto:** esta interacção desselecciona remotamente qualquer item seleccionado actualmente. Incorporar interacções remotas no protótipo reduz a fadiga do braço, o que poderia acontecer com o uso prolongado de interacções em que seja necessário apontar. Na Figura 5 é possível observar um exemplo de utilização da interacção *Touch & Interact*.

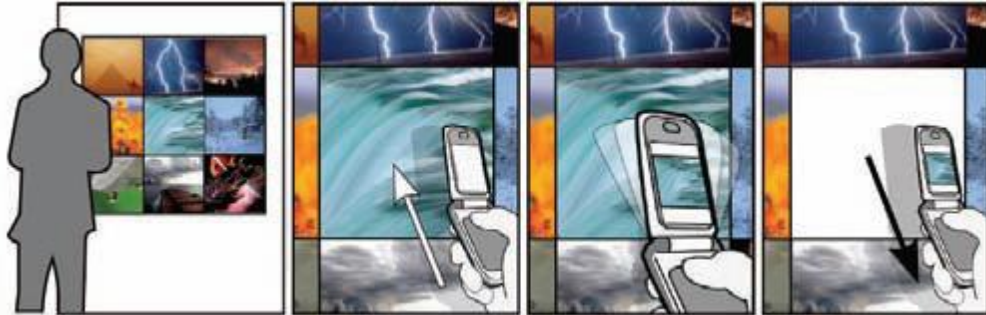


Figura 5 - *Touch & Interact*: o utilizador toca o ecrã com o telemóvel junto da foto que pretende seleccionar, e a foto é transferida do ecrã para o dispositivo móvel [11].

A interacção entre dispositivos móveis e ecrãs de grandes dimensões pode ser realizada utilizando fontes luminosas. O **C-Blink** [12] é um marcador de luz para ser utilizado em interacções com ecrãs de grandes dimensões que utiliza como fonte de luz visível o ecrã colorido de cristal líquido de um dispositivo móvel. Na Figura 6 é possível visualizar o sistema de teste para detectar sinais do *C-Blink*.



Figura 6 - Sistema de teste para detectar sinais do *C-Blink* [12].

Utiliza uma abordagem de diferença de tonalidades para implementar a sinalização luminosa. Os padrões resultantes raramente se repetem pelo que é um método de detecção confiável. A selecção de itens e a identificação do dispositivo móvel são efectuadas opticamente, e apenas é necessário um sensor de imagem por ecrã.

A interacção **Flashlight** [13] consiste na utilização do flash do telemóvel que é detectado por uma câmara colocada na frente de um ecrã, permitindo ao utilizador controlar o cursor do ecrã através dos movimentos efectuados com o telemóvel, como é visível na Figura 7.



Figura 7 – *Flashlight*: Uma câmara debaixo do ecrã segue os movimentos do flash [13].

Esta interacção é similar à utilizada no *C-Blink*, com a vantagem que na interacção *Flashlight* o ecrã não é usado como fonte de luz, e o ecrã continua a estar disponível para outra forma de interacções, como mostrar dados. Na verdade, todas as funcionalidades do telemóvel excepto o flash são mantidas para diferentes cenários de utilização, tornando esta interacção facilmente extensível. Além disso, o feixe de luz brilhante do flash da câmara do telemóvel permite uma interacção precisa mesmo à distância. Como esta interacção baseia-se principalmente em mover uma fonte de luz, pode ser facilmente utilizada em conjunto com ponteiros laser ou outras fontes de luz numa configuração multi-dispositivo.

O nível de interacção proporcionado entre utilizadores é também um aspecto importante na interacção entre dispositivos móveis e ecrãs de grandes dimensões. Na Universidade de *Milano-Bicocca*, alguns estudantes e professores utilizaram os seus

telemóveis para interagirem com 2 *SmartBoards* [14]. Podiam existir vários ecrãs na mesma sala, e os utilizadores podiam escolher o ecrã com que queriam interagir, tendo apenas que apontar o seu telemóvel para o ecrã desejado. Os utilizadores podiam enviar conteúdos (imagens, *clips* de vídeo, texto) para ser apresentados no ecrã, assim como controlar aplicações. Os telemóveis utilizados tinham capacidade de comunicação através de infra-vermelhos, como por exemplo os telemóveis *Nokia* da Série 60. A transmissão por infra-vermelhos ajustava-se à necessidade de permitir uma interacção direccional, já que os telemóveis podiam ser apontados, e a transmissão de curto alcance (aproximadamente 1 metro) permite uma desambiguação de qual o telemóvel que está a apontar para o ecrã.

Duas aplicações foram desenvolvidas e testadas, permitindo: uma interacção individual, onde se executava transferência de imagens e controlo de *slide-shows*, para os professores projectarem os seus slides numa aula; uma interacção colaborativa, tendo sido desenvolvida uma versão do Jogo do Galo, de modo a diferentes pessoas poderem interagir (os utilizadores podiam ser identificados através dos seus dispositivos, podendo operar em turnos). Na Figura 8 é possível observar um dos utilizadores a interagir com um dos *SmartBoards*.



Figura 8 - Utilizador a interagir com um dos *SmartBoards* [14].

O *TiltRacer*[15] é um jogo desenvolvido para ser utilizado tendo como dispositivo de entrada um telemóvel. Foram utilizados os sensores de movimentos do telemóvel *Nokia* 5500 para controlar o jogo. Este foi implementado numa versão *multiplayer* de modo a ser jogado num ecrã de grandes dimensões. A comunicação efectuou-se por *Bluetooth*, como se pode observar na Figura 9. Como se pode visualizar na Figura 10, apesar deste jogo

poder ser considerado um exemplo bastante simples em termos de gráficos e complexidade, este possui um alto nível de interacção com os utilizadores associado, proporcionando uma experiência social agradável.

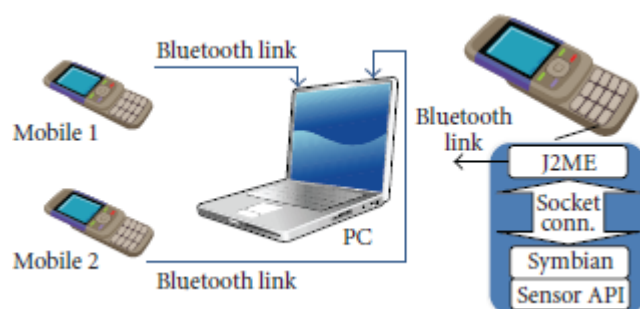


Figura 9 - Arquitectura de comunicação implementada no *TiltRacer* [15].



Figura 10 - *TiltRacer* jogado por utilizadores [15].

O *Doodle Space* [16] é uma aplicação interactiva desenvolvida para ecrãs de grandes dimensões, que permite a vários utilizadores pintarem em conjunto curvas 3D utilizando a câmara de dispositivos móveis. É utilizado um algoritmo baseado na sequência da imagem para estimar os parâmetros do movimento efectuado pelo telemóvel. As pessoas podiam não só desenhar as curvas, como também arrastá-las, rodá-las e aumentá-las de tamanho. A comunicação entre os telemóveis e o ecrã foi implementada através de *Bluetooth*. Na Figura 11 podemos verificar a arquitectura de comunicação do *Doodle Space*. Foram efectuados testes com utilizadores para testemunhar e aumentar a viabilidade e entretenimento do *Doodle Space*, como podemos visualizar na Figura 12. As principais

conclusões retiradas foram que o *Doodle Space* é capaz de reconhecer e monitorizar com confiança movimentos através de dispositivos móveis com câmaras, sendo que esta aplicação deverá ser destinada a utilizadores com experiência com dispositivos móveis. Contudo, necessita ainda de alguns melhoramentos para satisfazer a necessidade de designers profissionais.

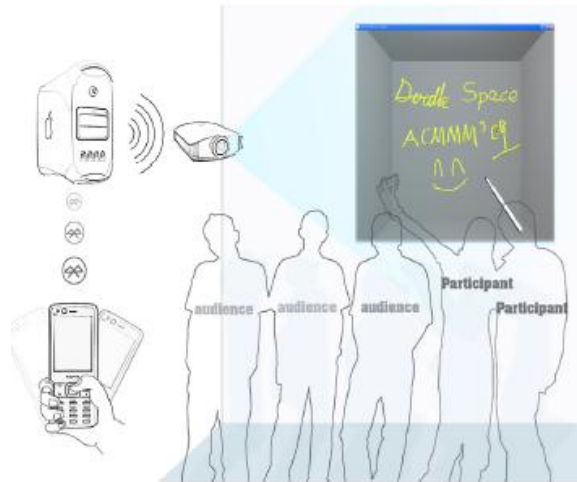


Figura 11 - Arquitectura de comunicação do *Doodle Space* [16].

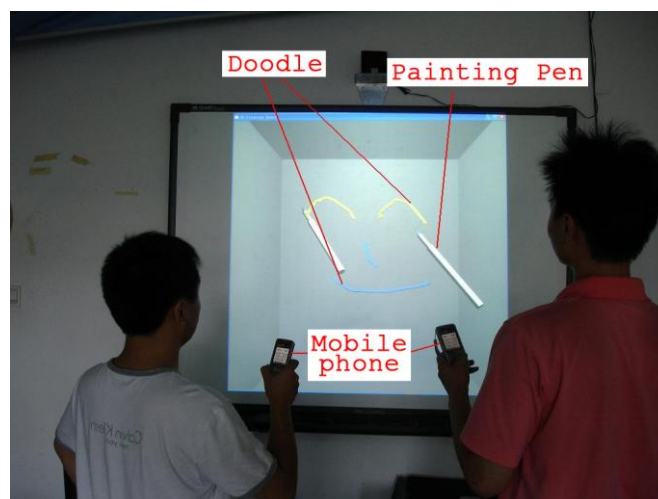


Figura 12 - *Doodle Space* testado por utilizadores [16].

Uma das tecnologias por explorar nos dispositivos móveis é o acelerómetro. Actualmente, este tipo de sensores ainda tem pouca importância no contexto de utilização dos telemóveis, sendo usados essencialmente para visualizar imagens e fazer *scroll* de

documentos. O facto de terem sido recentemente implementados em dispositivos faz com que o mercado ainda não esteja totalmente adaptado às suas características, o que é comprovado pelo facto do design dos dispositivos ainda não estar completamente adaptado à utilização de acelerómetros. É ainda necessário um maior estudo sobre a utilização destes sensores para poder determinar com maior exactidão quais os níveis de precisão adequados à sua utilização nos diferentes contextos de utilização. Um estudo [17] procurou averiguar as potencialidades de três tipos de movimentos possíveis de efectuar com o dispositivo móvel na mão, como ilustrado na Figura 13.

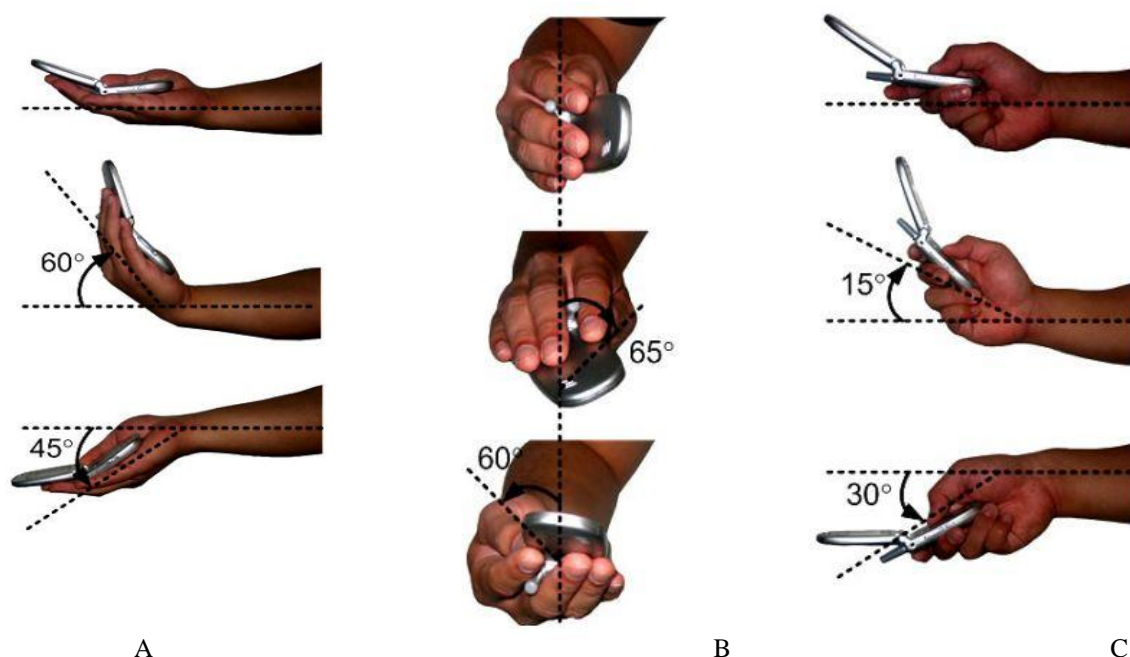


Figura 13 - Movimento de Flexão/Extensão (figura A), Pronação/Supinação (figura B) e Desvio Ulnar/Radial (figura C) [17].

Foram realizadas duas experiências para avaliar o desempenho do esforço efectuado pelo pulso neste tipo de movimentos. As principais conclusões das experiências foram que o eixo primário do acelerómetro deve ser constituído por movimentos de pronação/supinação, para realizar selecções para a esquerda/direita, e de flexão/extensão para realizar selecções para cima/baixo.

Foi possível verificar que uma interacção entre dispositivos móveis e ecrãs de grandes dimensões é caracterizada pelas formas de interacção disponíveis, o tipo de comunicação implementado e o nível de interacção proporcionado aos utilizadores.

3. Ferramentas Utilizadas

Neste capítulo é apresentado o sistema operativo móvel *Android*, assim como são aprofundadas as suas características e potencialidades. São também introduzidas as ferramentas utilizadas ao longo do trabalho efectuado.

3.1. *Android*

A escolha do sistema operativo móvel recaiu no *Android*, visto que é considerado pela comunidade científica como uma das tecnologias emergentes mais promissoras na área da computação móvel [18]. Uma das suas principais vantagens é a sua flexibilidade, comprovada pelo facto de ser uma plataforma totalmente personalizável. O facto desta plataforma móvel ser *open-source* possui uma série de vantagens, tais como o seu potencial de disseminação elevado, já que o livre acesso ao hardware dos dispositivos móveis que possuem *Android* vai permitir o aparecimento de diversas aplicações gratuitas, assim como diversa ajuda para conceber novas aplicações. Devido a estes factores, os dispositivos *Android* tendem a ser mais económicos, o que poderá aumentar a sua procura e torná-lo uma das principais plataformas móveis do mercado a curto prazo. Neste trabalho, foi usada a versão 1.6 do sistema operativo, a última versão disponível no momento em que foi iniciado o projecto.

3.1.1. Descrição Geral

A evolução da tecnologia possibilitou o aumento das capacidades dos dispositivos móveis, permitindo funcionalidades como a utilização do *GPS*, a navegação na Internet ou chamadas com vídeo. Assim, um dos componentes mais importantes neste tipo de dispositivos é o sistema operativo.

O *Android* é um sistema operativo para dispositivos móveis, que foi lançado no mercado em Novembro de 2007 e inicialmente desenvolvido pela Google e posteriormente pela *Open Handset Alliance*, um conjunto de 65 empresas que desenvolve software *open-source* para dispositivos móveis, das quais fazem parte a *Google*, *Qualcomm*, *LG*, *T-Mobile*, *Nvidia*, *Wind River Systems*, *Motorola*, *Intel*, *Vodafone*, *HTC*, *Samsung*, *Texas Instruments*, entre outras.

Esta plataforma diferencia-se das restantes pelo facto de ser a primeira plataforma para dispositivos móveis verdadeiramente aberta e pela liberdade e flexibilidade que oferece à indústria de dispositivos móveis, programadores e utilizadores.

O *Android* possui como principais características:

- Um quadro de aplicações que permite a reutilização e substituição dos componentes;
- Uma máquina virtual *Dalvik* otimizada para dispositivos móveis;
- Um *browser* integrado baseado no motor *WebKit open-source*;
- Gráficos otimizados alimentados por uma biblioteca própria de gráficos 2D e uma biblioteca de gráficos 3D baseada nas especificações de *OpenGL ES 1.0*;
- Base de dados *SQLite* para armazenamento de dados;
- Suporte de multimédia para áudio, vídeo e formatos de imagem *MPEG4*, *H.264*, *MP3*, *AAC*, *AMR*, *JPG*, *PNG* e *GIF*;
- Telefonia *GSM*;
- *Bluetooth*, *EDGE*, *3G* e *Wi-Fi*;
- Suporte para câmara de vídeo e fotográfica, *GPS*, bússola e acelerómetro;
- Ambiente de desenvolvimento completo que inclui um emulador do dispositivo, ferramentas para *debugging*, memória e perfis de desempenho, e um *plugin* para o Eclipse IDE.

O facto de todas as aplicações estarem profundamente integradas com o sistema, permite aos programadores terem acesso a todas as *APIs* da plataforma, da mesma forma que têm acesso às aplicações nucleares, como por exemplo o envio de mensagens ou a realização de telefonemas.

Arquitectura da Plataforma *Android*

Como se pode ver na Figura 14, a arquitectura desta plataforma consiste em quatro camadas: o *kernel*, as bibliotecas, o *framework* e as aplicações [19].

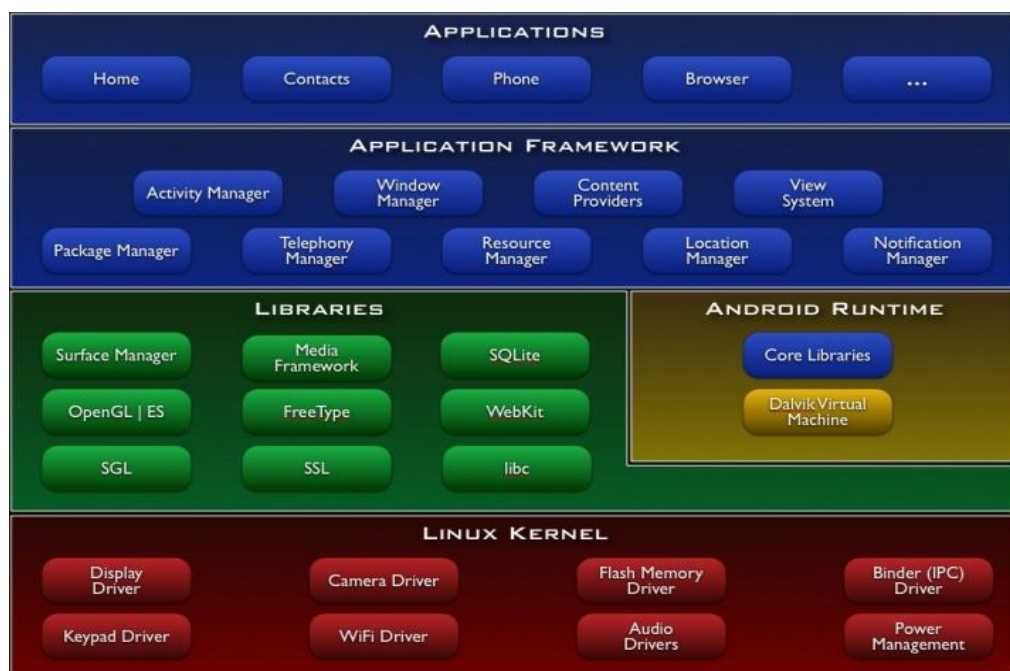


Figura 14 - Representação esquemática da arquitectura da plataforma *Android* [19].

O *kernel* é a base da plataforma sobre a qual corre tudo o resto, e que funciona como camada de abstracção entre o hardware e as restantes camadas. Consiste num kernel *linux* 2.6, sendo que nesta camada são executadas os serviços de gestão de memória e de processos, assim como os drivers necessários para instalar o *Android* nos vários dispositivos.

Na camada das bibliotecas, que inclui as bibliotecas C e C++ usadas pelos componentes do *Android* e as bibliotecas Java, encontra-se a máquina virtual *Dalvik*,

principal componente do *Android Runtime*. Esta máquina virtual foi desenhada especificamente para correr no ambiente limitado proporcionado pelos dispositivos móveis, cujos recursos, como CPU, bateria e capacidade de armazenamento de dados são reduzidos, e para permitir que várias instâncias da máquina corram ao mesmo tempo.

A *framework*, escrita em Java, foi desenhada de modo a simplificar a reutilização de componentes, por parte das aplicações. Oferece à camada superior, as aplicações, um conjunto de serviços como por exemplo: Vistas, que incluem listas, grelhas, caixas de texto ou botões, gestor de recursos e de actividades, que são estruturas base das aplicações *Android* e são explicadas mais à frente. Esta camada pode ser vista como um conjunto básico de ferramentas que permite ao utilizador criar aplicações ferramentas mais complexas.

A última camada corresponde às aplicações que, apesar de escritas em Java, possuem uma estrutura um pouco diferente das aplicações não *Android*. Todas as aplicações têm acesso aos mesmos recursos fornecidos pela camada inferior, independentemente de fazerem parte do núcleo da plataforma ou terem sido desenvolvidas por terceiros.

Estrutura das Aplicações *Android*

Uma aplicação *Android* consiste num arquivo *.apk* com código Java compilado e ficheiros externos (ficheiros que não contém código, como imagens ou ficheiros XML), e corre no seu próprio processo Linux, com a sua própria máquina virtual. A gestão dos processos não é da responsabilidade da aplicação mas sim do sistema operativo *Android*, que cria um processo assim que uma nova aplicação é lançada e destrói-o assim que os recursos do sistema são precisos para outras aplicações. A escolha do processo a destruir baseia-se nos seguintes níveis de prioridade, apresentados por ordem decrescente de importância:

- **Processo em primeiro plano**, que é aquele que é necessário para suportar uma acção que o utilizador esteja a realizar num dado momento;
- **Processo visível**, é um processo que, embora não suporte nenhum componente que esteja acessível por parte do utilizador, contém componentes que ainda são visíveis;

- **Processo serviço**, é um processo que suporta um serviço que não se encontra nas duas categorias anteriores;

- **Processo em background**, é um processo que suporta uma actividade que não se encontra visível;

- **Processo vazio**, é um processo que não suporta qualquer componente.

Existem quatro blocos fundamentais para a construção de aplicações *Android*: *Activities*, *Services*, *Broadcast Receivers* e *Content Providers*, não sendo contudo obrigatório que uma aplicação seja formada por todos eles [20].

As *Activities* consistem numa única classe e correspondem a um ecrã. São responsáveis por apresentar a interface gráfica e gerir os eventos que aí ocorram. Esta é uma das principais diferenças da arquitectura das aplicações *Android* para as aplicações Java dos demais sistemas, a existência de vários pontos de entrada proporcionadas pelas *Activities*, por oposição a um único ponto de entrada proporcionado pelo método *main*.

Um exemplo vulgarmente utilizado para exemplificar a relação entre *Activities* e aplicações é o exemplo de uma aplicação para gerir mensagens escritas, que poderia ser composta por três *Activities*: uma *Activity* para apresentar todas as mensagens na caixa de entrada assim como as opções relativas às acções do utilizador, uma *Activity* para apresentar a mensagem seleccionada pelo utilizador na *Activity* anterior e uma *Activity* para apresentar o formulário utilizado pelo utilizador para criar uma nova mensagem e enviá-la. Assim, uma aplicação pode ser definida como uma pilha de *Activities* e, sempre que o utilizador navega para um novo ecrã, a *Activity* correspondente a esse ecrã é adicionada ao topo da pilha. Caso volte ao ecrã anterior, a *Activity* que se encontra no topo da pilha é removida.

Os *Services* não apresentam interface gráfica, correm em *background* por um período de tempo indefinido e podem comunicar com as *Activities*. São particularmente úteis para manter em execução aplicações ou processos enquanto outros são lançados, como por exemplo, manter um leitor de música activo enquanto outras aplicações são executadas.

Os *Broadcast Receivers* são responsáveis quer pela recepção de anúncios referentes a eventos externos, como por exemplo a chegada de uma mensagem ou uma notificação de que outra aplicação terminou um *download*, quer pelas acções de resposta a esses

anúncios, que podem consistir por exemplo em avisos sonoros, luminosos ou a inclusão de ícones na barra de estado.

Os **Content Providers** disponibilizam os recursos exclusivos de uma aplicação a outras aplicações, permitindo assim que estas armazenem e acessem aos dados e os manipulem, através de qualquer mecanismo de registo de dados, como por exemplo bases de dados *SQLite* ou dados em ficheiros. Este componente é a única forma de partilhar dados uma vez que não existe uma única área comum de armazenamento à qual todos os pacotes do *Android* tenham acesso.

Todos os componentes podem comunicar entre si, enviando mensagens denominadas **Intents** [21], que são objectos da classe *Intent* e que contém uma descrição da operação a ser realizada. Podem ser usados para lançar *Activities*, para enviar dados ou notificações a outros componentes que os capturam através dos *Broadcast Receivers* ou lançar serviços e conectá-los a *Activities*.

Os dois atributos principais numa estrutura *Intent* de dados são a acção, que consiste numa *string* que indica qual a acção a ser realizada, e os dados, que são expressos através de um *Uri* e que indicam quais os dados a ser utilizados pela acção. Além destes atributos, existem ainda atributos secundários como os *Extras*, que permitem o envio de dados adicionais necessários para realizar a acção. Os *Intents* podem ser de dois tipos: nos Explícitos, é especificado qual o componente a ser lançado e assim é indicada a classe que deve correr para executar esse componente e nos Implícitos, não é especificado nenhum componente, mas contém informação suficiente para que o sistema possa determinar qual dos componentes disponíveis melhor se adequa aos requisitos impostos pelo *Intent*. Essa escolha é feita mediante os valores do *Intent* e um conjunto de filtros que os componentes têm, que consistem numa descrição dos valores que esse componente pode receber. No caso de existir mais de que uma aplicação com componentes capazes de lidar com o conjunto de valores presentes no *Intent*, é o utilizador que escolhe qual a aplicação a utilizar. Esses filtros são descritos num ficheiro XML, presente em todas as aplicações, denominado **AndroidManifest** que, além dos *Intent Filters*, contém uma descrição de todos os componentes que compõe a aplicação assim como das permissões entre outras características da aplicação. Resumindo, o *AndroidManifest* contém a informação essencial para que o sistema possa executar a aplicação.

Outra característica das aplicações escritas para *Android* é a possibilidade de criar as **Interfaces Gráficas** de duas maneiras distintas, através de código Java ou através de ficheiros XML, sendo que esta última forma permite separar as interfaces gráficas da parte lógica da aplicação, e assim desenhar as aplicações segundo o princípio MVC (*Model-View-Controller*). As interfaces consistem em objectos das classes *View* e *ViewGroup*, como se pode observar na Figura 15. *View* é a unidade básica dos elementos gráficos, responsável pelos atributos dos elementos gráficos, como medidas ou mudanças de foco, assim como por se desenhar no ecrã. Os *ViewGroup* são um tipo especial de *Views* que permite agrupar conjuntos de *Views* e elementos *ViewGroup* numa estrutura em árvore e são responsáveis por desenhar os elementos que contêm, permitindo assim adicionar elementos gráficos complexos como uma única entidade.

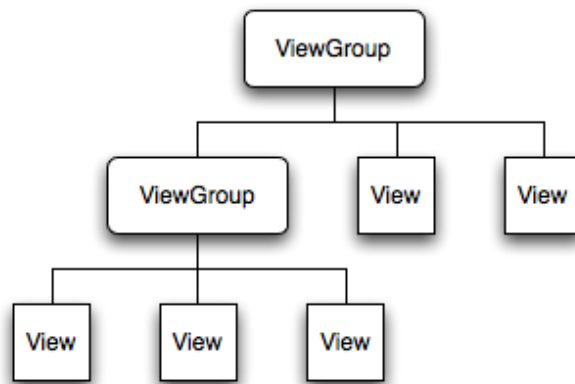


Figura 15 - Estrutura das interfaces gráficas nas aplicações *Android* [22].

Os dois modos de criar as interfaces gráficas não são mutuamente exclusivos, sendo a *framework* do *Android* suficientemente flexível e robusta para permitir que se adicionem, removam ou alterem elementos gráficos declarados no ficheiro XML através do código Java.

Os **ficheiros XML** com o *layout*, não são no entanto os únicos ficheiros externos a poderem ser incluídos numa aplicação *Android*. Ficheiros com imagens, valores (como *strings* por exemplo) ou animações podem também ser incluídos nas aplicações *Android*. Estes ficheiros encontram-se organizados pelo tipo de recurso a que pertencem existindo seis tipos:

- *anim* para animações;

- *drawable* para imagens;
- *layout* para interfaces gráficas;
- *values* para definições de valores como *strings* ou *arrays*;
- *xml* para ficheiros XML;
- *raw* para os ficheiros que serão directamente copiados para o dispositivo sem serem compilados ao contrário dos ficheiros nas restantes pastas, que ou são compiladas para binário ou comprimidas para formatos mais eficientes.

Os nomes dos directórios são ainda responsáveis, além da ligação entre os recursos e o seu tipo, pela ligação destes às definições do sistema (como a língua e orientação do dispositivo), através de *strings* que indicam o valor da definição a que correspondem e que são adicionadas no fim do nome do directório.

Por exemplo, para construir uma aplicação com suporte para português e inglês, devem existir os directórios *values-pt* e *values-en*, uma vez que é no directório *values* que se encontram as definições das *string* e os sufixos, pt e en, definem a linguagem. Vários valores podem ser combinados, separando as *strings* correspondentes através de hífenos.

Todos os recursos devem ter um identificador único, de modo a serem identificados e acedidos através do código ou de outros recursos.

Android Market

Um dos aspectos mais atractivos do *Android* é o facto de qualquer programador poder criar e disponibilizar as suas aplicações. O *Android Market* é a infra-estrutura que estabelece a ligação entre utilizadores e programadores, para que seja fácil e atractivo aos utilizadores descarregá-las e instalá-las nos seus dispositivos. Para disponibilizar as suas aplicações, os programadores apenas necessitam de estar registados na comunidade *Android* e fazer *upload* da sua aplicação, à qual podem associar uma descrição. As aplicações não são submetidas a qualquer tipo de validação, estando no entanto sujeitas aos termos das operadoras e às regras da Google em relação ao conteúdo das aplicações [23] (por exemplo não são permitidas aplicações com alusões a comportamento violento ou que não respeitem direitos de autor), pelo que esta pode remover remotamente as aplicações

nos dispositivos além de as retirar do *Android Market*.

A esmagadora maioria das aplicações é gratuita, mas também é possível disponibilizar aplicações pagas, em que o autor da aplicação tem direito a 70% das receitas, e os restantes 30% são para as operadoras e para despesas relacionadas com o alojamento [24].

Esta forma de distribuir aplicações assemelha-se à utilizada pela *Apple* para distribuir aplicações para o seu dispositivo móvel, o *iPhone*, apesar de esta submeter todas as aplicações a um processo de validação e só então, caso sejam validadas, podem ser distribuídas.

Esta infra-estrutura é um dos elementos mais importantes de todo o universo em torno do *Android*, e o seu sucesso poderá determinar o sucesso da plataforma.

3.1.2. Software Android

Android SDK

O *Android SDK* [25] é um conjunto de ferramentas destinado à criação de aplicações para *Android* (quer como *plugin* do *Eclipse* quer de forma autónoma). Além das bibliotecas e da máquina virtual *Dalvik*, oferece aos programadores a documentação sobre *APIs*, tutoriais, aplicações exemplo e o seu respectivo código, e várias ferramentas bastante úteis no desenvolvimento de aplicações. Algumas dessas ferramentas são o emulador do dispositivo, uma ferramenta que permite criar um cartão de memória virtual que simula um cartão de memória (*SD Card*), e um monitor da máquina virtual *Dalvik*, que permite analisar os recursos consumidos pela aplicação como a memória ou *CPU* utilizados ou ainda ver o estado da *Heap* da máquina virtual.

O emulador é um dispositivo móvel virtual que simula as características, quer de *hardware* quer de *software*, do dispositivo real, e é extremamente útil no desenvolvimento de aplicações, não só em situações em que o programador ainda não tem um dispositivo real, mas também quando pretende criar aplicações que promovam a interacção entre vários dispositivos, uma vez que é possível ter várias instâncias do emulador a correrem. Permite simular, além de todas as funcionalidades (com excepção da realização de chamadas),

algumas condições que podem afectar o rendimento das aplicações (como a latência ou velocidade da rede) e o comportamento dos utilizadores, como a sua localização ou percursos efectuados.

A interacção com o utilizador é feita através de uma interface que consiste numa representação gráfica do dispositivo, que simula o ecrã táctil e todos os botões do dispositivo, como se pode ver na Figura 16.

Esta interface é configurável, permitindo alterar o dispositivo que está a ser usado e a sua orientação (*portrait* ou *landscape*). Desta forma é possível testar as aplicações para diferentes dispositivos, com diferentes medidas de ecrã e para diferentes orientações.

O *Android SDK* foi usado com o intuito de emular um dispositivo móvel *Android*, de modo a não ser necessário recorrer ao dispositivo real numa fase inicial do trabalho.

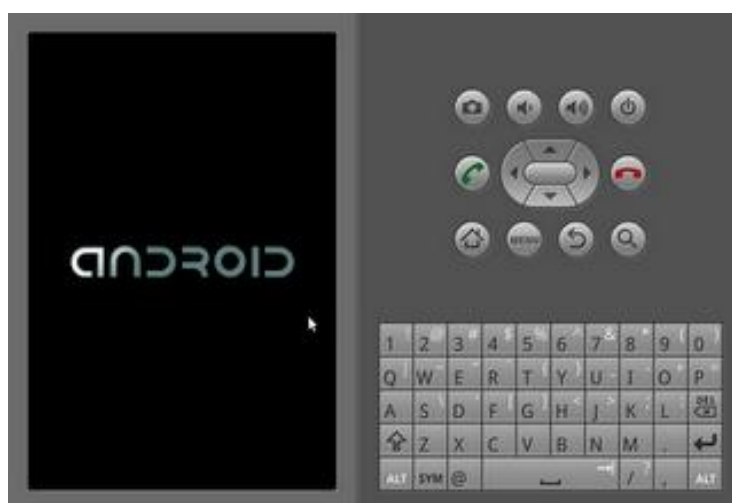


Figura 16 - Interface gráfica do emulador do *SDK Android* [25].

Sensor Simulator

O *Sensor Simulator* [26] é uma aplicação que permite simular em tempo real os dados dos sensores do dispositivo real. Os dados são controlados pelo utilizador através da interface gráfica da aplicação, que além de apresentar as configurações, apresenta também uma representação tridimensional do dispositivo que reflecte as alterações nos valores de orientação introduzidos pelo utilizador. Estes valores são introduzidos através de três barras horizontais, sendo cada uma delas, responsável pelo controle da rotação em torno de cada um dos eixos de um referencial tridimensional, como se pode ver na Figura 17.

Esta aplicação funciona como um servidor, que escuta os pedidos dos clientes, as aplicações *Android*, numa porta definida pelo utilizador, aos quais responde com um fluxo constante de dados, que são gerados de acordo com as acções do utilizador.

O *Sensor Simulator* foi usado para simular os dados provenientes do acelerómetro e da bússola digital, de modo a não ser necessária a utilização do dispositivo real numa fase inicial do trabalho.

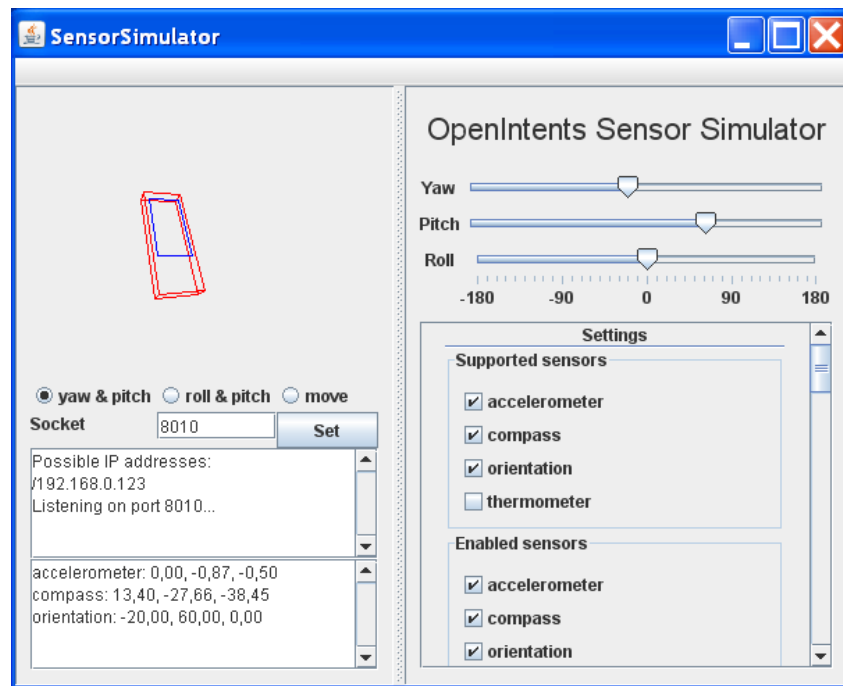


Figura 17 - Interface gráfica do *Sensor Simulator* [26].

3.1.3. Hardware Android

Dispositivo Móvel utilizado

O dispositivo móvel utilizado ao longo do projecto foi o *HTC Dream* [27] (ou *T-Mobile G1*), o primeiro dispositivo a incluir a plataforma *Android* e que foi desenvolvido pela HTC, uma das empresas que fundou a *Open Handser Alliance*. Foi lançado em Outubro de 2008 em duas versões, comercial e de desenvolvimento, sendo que a segunda consiste numa versão desbloqueada, disponível apenas para membros registados na comunidade *Android*, e que permite ao utilizador ter acesso completo aos ficheiros internos do dispositivo com permissões de super-utilizador. Do seu *design* destacam-se o ecrã táctil

de 8,1 centímetros com uma resolução 320x840, sob o qual se encontra um teclado deslizante (*QWERTY*), e a secção inferior (característica dos dispositivos da *HTC* com o *Android*), na qual se encontram os 5 botões de navegação e a *trackball*. Através deles o utilizador tem acesso às funções de telefonia, voltar ao ecrã anterior, voltar ao ambiente de trabalho, aceder ao menu ou navegar pelos itens de um ecrã. Possui ainda mais dois conjuntos de controlos laterais, que controlam o volume e o acesso à câmara de 3.2 MegaPixéis.



Figura 18 - *HTC Dream* [27].

Como se trata de um dispositivo móvel, as tecnologias de transferência de dados sem fios são fundamentais para o seu sucesso. Para um dispositivo móvel com *Android* são-no ainda mais, uma vez que é uma plataforma projectada para “viver” na rede, para aceder e interagir com uma grande quantidade de conteúdos remotos. Assim, este dispositivo permite o acesso a redes 3G (com velocidades de 7,2Mbps para *download* e de 2Mbps para *upload*), a redes *Wi-Fi* e oferece ainda suporte para *bluetooth*.

Do ponto de vista dos programadores, duas das características mais interessantes são o conjunto de sensores (acelerómetro, bússola e GPS), que permitem desenvolver aplicações com uma componente ubíqua bastante forte, e a possibilidade de ligar o dispositivo a um computador pessoal através de um mini cabo USB. Esta característica é bastante importante na medida em que permite a quem está a desenvolver aplicações, aceder aos ficheiros internos do *Android* e fazer *debugging* da sua aplicação.

Sensores

Acelerómetro

A aceleração é definida como a taxa de mudança de velocidade: assim, os acelerómetros medem a rapidez com que a velocidade do dispositivo varia numa determinada direcção. Usando um acelerómetro, é possível detectar movimento, assim como calcular a taxa de variação de velocidade desse movimento.

A aceleração pode ser medida em três eixos direccionais: para a frente e para trás (longitudinal), para a esquerda e para a direita (lateral), e para cima e para baixo (vertical):

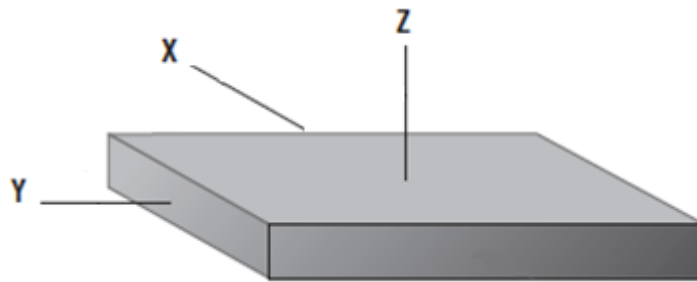


Figura 19 - Orientação dos eixos do acelerómetro do dispositivo móvel [28].

- **Vertical** – Aceleração ascendente ou descendente: a deslocação do dispositivo para cima representa o movimento positivo, como quando está a ser levantado, e o movimento do dispositivo para baixo representa o movimento negativo;
- **Longitudinal** – Aceleração para a frente ou para trás: o movimento é positivo quando a aceleração é feita para a frente e negativo quando o movimento é feito para trás;
- **Lateral** – Aceleração para a esquerda e para a direita: o movimento é positivo quando o dispositivo se desloca para a direita e negativo quando o dispositivo se desloca para a esquerda [28].

Os acelerómetros são incapazes de distinguir entre aceleração da gravidade e aceleração devido ao movimento. Como consequência, um acelerómetro vai detectar aceleração no eixo Z de -9.8 m/s^2 quando se encontra em repouso.

O acelerómetro presente no dispositivo móvel utilizado para realização desta dissertação é o chip *Bosch Sensortec BMA150* de três eixos, sendo a *refresh rate* indicada no *Datasheet* de 3 kHz [29].

Bússola Digital

Os sensores de orientação são uma combinação da bússola digital que fornece o *yaw* e do acelerómetro que ajudam a determinar o *pitch* e o *roll*. De notar que a plataforma *Android* calcula automaticamente a orientação do dispositivo com base nos valores do acelerómetro ao longo dos três eixos.

A orientação do dispositivo é descrita ao longo dos três eixos:

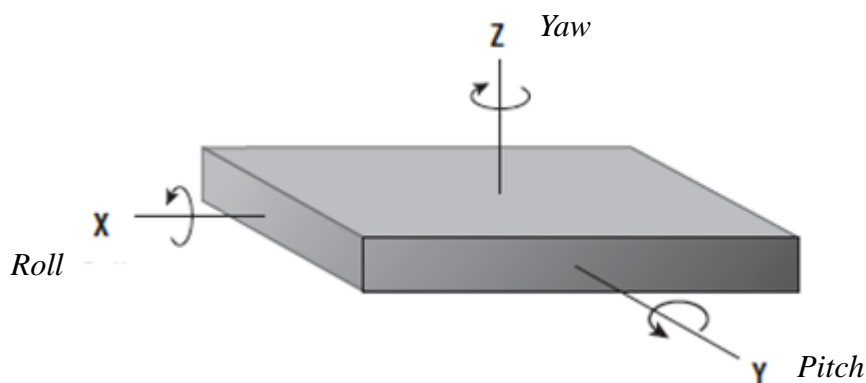


Figura 20 - Orientação dos eixos da bússola digital do dispositivo móvel [28].

- ***Yaw*** – É o movimento efectuado em torno do eixo Z, onde $0^{\circ}/360^{\circ}$ corresponde à direcção Norte, 90° à direcção Este, 180° à direcção Sul e 270° à direcção Oeste;

- ***Pitch*** – É o movimento realizado em torno do eixo Y, onde 0° indica que o dispositivo está virado para cima, -90° indica que o dispositivo está em pé (com a “cabeça” virada para cima), 90° quando está com a “cabeça” virada para baixo, e $180^{\circ}/-180^{\circ}$ quando o dispositivo está virado para baixo;

- ***Roll*** – É o movimento executado em torno do eixo X, podendo variar entre 90° e -90° . Quando é indicado 0° o dispositivo está com a sua frente virada para cima ou para baixo, -90° o dispositivo está com a sua frente inclinada para a esquerda e 90° o dispositivo está com a sua frente inclinada para a direita.

O dispositivo é considerado em repouso (*yaw*, *pitch* e *roll* a 0°) quando o lado do telemóvel onde está o ecrã se encontra virado para cima [28].

A bússola digital presente no dispositivo móvel utilizado para realização desta dissertação é o modelo de três eixos AK8973 da *Asahi Kasei* [30].

3.2. *Eclipse*

O *Eclipse* [31] é um Ambiente Integrado de Desenvolvimento (IDE) com suporte para várias linguagens, entre as quais o Java, sendo que para esta, é já o IDE mais utilizado em todo o mundo. Com uma arquitectura baseada em *plugins*, permite que novas funcionalidades sejam adicionadas de forma a aproximar a plataforma das necessidades do utilizador. Foi esta arquitectura que permitiu utilizar o *Eclipse* como uma das principais ferramentas de desenvolvimento de software neste projecto, uma vez que foi criado um *plugin* para desenvolver aplicações para *Android* através do *Eclipse*.

Esta plataforma foi utilizada para realizar algumas aplicações desenvolvidas em Java ao longo do projecto.

3.3. *Visual Studio*

Microsoft Visual Studio [32] é um Ambiente Integrado de Desenvolvimento da *Microsoft*, e pode ser usado para desenvolver aplicações de consola e aplicações de interface gráfica de utilizador, através das aplicações *Windows Forms*, aplicações Web, e serviços Web utilizando código nativo juntamente com o código gerado para todas as plataformas suportadas pela *Microsoft Windows*, *Windows Mobile*, *Windows CE*, *.NET Framework*, *.NET Compact Framework* e *Microsoft Silverlight*.

Esta foi a plataforma usada para desenvolver o *DetiGuide*. A versão utilizada foi a *Microsoft Visual Studio 2008*, na linguagem C#.

3.4. *VTK*

O VTK (*Visualization ToolKit*) [33] é uma biblioteca aberta de classes em C++ que possui várias interfaces para linguagens interpretadas, tais como, *Tcl/Tk*, *Java*, *Python* e C#. Utiliza-se em computação gráfica 3D, no processamento de imagens e em visualizações.

VTK suporta uma ampla variedade de algoritmos de visualização, incluindo: vector escalar, tensor, textura, métodos volumétricos e técnicas avançadas de modelação, tais

como: modelagem implícita, redução de polígonos, suavização de malhas, corte de contorno, e triangulação de *Delaunay*. VTK possui uma estrutura de visualização da informação, tem um conjunto de *widgets* de interacção 3D, suporta o processamento paralelo, e integra-se com diversas bases de dados nas interfaces gráficas de utilizadores como *Qt* e *Tk*.

As bibliotecas de VTK foram utilizadas para realizar as aplicações gráficas realizadas nas primeiras experiências desenvolvidas para teste. Foi usada a versão 5.4.2, na linguagem Java.

A instalação de VTK no computador foi um processo longo e com bastantes dificuldades, pelo que se inclui em anexo um guia de instalação de modo a ajudar aqueles que necessitarem de o instalar (anexo C).

4. Aplicações Desenvolvidas

Neste capítulo é feita uma descrição da arquitectura de comunicação implementada, e também do software desenvolvido ao longo do projecto.

4.1. Arquitectura de Comunicação

A figura seguinte representa a comunicação implementada ao longo de todo o projecto.

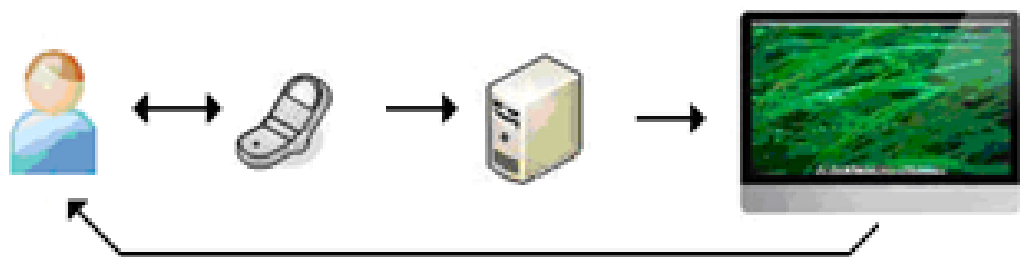


Figura 21 - Esquema da comunicação implementada ao longo do projecto [3].

A comunicação foi implementada através da arquitectura cliente - servidor, já que o *Bluetooth* não estava completamente integrado na plataforma *Android* na altura em que foi realizado o projecto. No entanto, já é possível utilizar o *Bluetooth* com a última versão da plataforma *Android* (versão 2.1). Inicialmente foi implementado o protocolo TCP/IP, um

dos principais protocolos utilizados na estruturação da Internet, que se caracteriza pela sua versatilidade e robustez, já que verifica se os dados são enviados de forma correcta, na ordem apropriada e sem erros. No anexo A é apresentada uma solução de implementação do cliente e servidor TCP/IP em Java.

No entanto, após testes iniciais constatou-se que este protocolo era inadequado devido às suas características causarem atrasos significativos na comunicação, já que este protocolo assegura a entrega de todos os valores enviados pelo dispositivo móvel. A solução idealizada para ultrapassar este problema foi utilizar um protocolo que causasse menos atrasos na comunicação, desprezando os valores acumulados para entrega, causadores dos atrasos. Um protocolo que verificou as características pretendidas foi o protocolo UDP, que é um protocolo não confiável, pois não assegura a entrega de todos os pacotes enviados, o que é benéfico para este caso específico. Apresenta também como desvantagem o facto de ser um protocolo orientado a fluxos de bytes sem início e sem fim, o que não permite a diferenciação dos pacotes recebidos em relação ao tamanho destes, causando uma maior facilidade de mistura do conteúdo de diferentes mensagens, proporcionando a introdução de caracteres residuais nos pacotes recebidos. No anexo B é apresentada a solução usada para o cliente e servidor UDP.

Na tabela 1 estão resumidas as principais diferenças entre o protocolo TCP/IP e o protocolo UDP [34].

Protocolo TCP/IP	Protocolo UDP
Confiável: O protocolo TCP é orientado para ligações. Quando uma mensagem é enviada será entregue a menos que a ligação falhe. Se a ligação for perdida, o servidor irá requerer a parte perdida. Não há corrupção durante a transferência da mensagem.	Não Confiável: O protocolo UDP é um protocolo sem ligação. Quando uma mensagem é enviada, não é possível determinar se a mensagem será entregue, já que é possível esta perder-se pelo caminho. Pode haver corrupção durante a transferência da mensagem.
Ordenado: Se forem enviadas duas mensagens ao longo de uma ligação, uma após a outra, sabe-se que a primeira mensagem enviada irá chegar ao outro lado primeiro. Não há preocupação da mensagem chegar na ordem errada.	Não Ordenado: Se forem enviadas duas mensagens, não é possível saber com que ordem estas irão chegar ao outro lado.

Pesado: Se as mensagens chegarem ao outro lado com a ordem errada, pedidos de reenvio são efectuados, e todas as partes da mensagem têm que ser colocadas juntas novamente, o que requer bastante trabalho para reunir as partes.	Leve: Sem ordem nas mensagens e sem controlo das ligações. A mensagem é apenas enviada sem verificações, o que torna este protocolo bastante rápido. A placa de rede e o sistema operativo também efectuam poucas operações para traduzir os dados dos pacotes.
Streaming: A mensagem é lida como uma <i>stream</i> , sem distinção onde um pacote acaba e outro começa. Pode haver vários pacotes numa recepção da mensagem.	Datagramas: Os pacotes são enviados individualmente e é garantido que chegam completos. Há apenas um pacote numa recepção da mensagem.
Exemplos de utilização: <i>World Wide Web</i> (<i>Apache TCP</i> porta 80), e-mail (<i>SMTP TCP</i> porta 25 <i>Postfix MTA</i>), <i>File Transfer Protocol</i> (<i>FTP</i> porta 21) e <i>Secure Shell</i> (<i>OpenSSH</i> porta 22) etc.	Exemplos de utilização: <i>Domain Name System</i> (<i>DNS UDP</i> porta 53), <i>streaming media applications</i> como <i>IPTV</i> ou filmes, <i>Voice over IP</i> (<i>VoIP</i>), <i>Trivial File Transfer Protocol</i> (<i>TFTP</i>) e jogos <i>multiplayer</i> online etc

Tabela 1 - Diferenças entre o protocolo TCP/IP e o protocolo UDP [34].

4.2. Testes Iniciais

As aplicações desenvolvidas ao longo deste projecto foram o controlo de um sólido, um jogo e o *DetiGuide*. Todas estas aplicações são controladas através dos movimentos efectuados com o dispositivo móvel *Android*. O controlo do sólido permitiu aprender a usar e a aceder aos diversos sensores do dispositivo móvel utilizado, e também testar a arquitectura de comunicação implementada, especificamente a velocidade e a fiabilidade da comunicação. Numa fase posterior, a criação do jogo teve como objectivo testar o potencial do acelerómetro para interacção. Por último, o propósito do *DetiGuide* foi testar diferentes formas de interacção possíveis usando um dispositivo móvel.

Foi utilizado o Eclipse com linguagem Java para desenvolver o código para o dispositivo móvel, assim como para o servidor. Como a única linguagem disponível para programar em *Android* é em Java, optou-se por implementar o servidor também em Java. Para gerar o sólido e os outros objectos utilizados no jogo foram utilizadas as bibliotecas VTK, utilizando também a linguagem Java, por uma questão de coerência com as escolhas anteriores.

4.2.1. Controlo de um Sólido no Ecrã

Para realizar esta aplicação, os testes realizados inicialmente foram efectuados usando um simulador dos sensores do dispositivo móvel, o *Sensor Simulator*, o que permitiu maior controlo sobre os valores obtidos. No anexo D é disponibilizado o código desenvolvido para aceder aos valores dos sensores.

Existiam 2 opções para tentar sincronizar o sólido com os movimentos do telemóvel: aplicar directamente no sólido os ângulos de *Euler*, *yaw*, *pitch* e *roll*, e utilizar matrizes de rotação para descrever o movimento do dispositivo móvel. Numa primeira fase, tentou-se usar a solução mais simples, aplicando os ângulos de *Euler* directamente no sólido. Após várias tentativas, constatou-se que não se conseguia obter uma sincronização a três dimensões utilizando apenas estes três valores, devido ao *Gimbal Lock*.

O *Gimbal Lock* [35] é a perda de um grau de liberdade quando dois dos três eixos espaciais se sobrepõem e não conseguem compensar a rotação sobre um dos eixos tridimensionais, como é ilustrado na Figura 22 e 23. Este problema foi ultrapassado utilizando matrizes de rotação para descrever o movimento do dispositivo móvel. O próprio dispositivo móvel calcula internamente uma matriz de rotação, utilizando os valores do acelerómetro e da bússola digital, que o VTK, biblioteca de interfaces gráfica utilizada para este projecto, aceita directamente como método de entrada dos valores oriundos do telemóvel. No anexo E poderá ser consultado o código implementado que resolveu o problema do *Gimbal Lock* através de matrizes de rotação. Na figura 24 podemos visualizar o resultado final da sincronização entre o dispositivo móvel e o sólido no computador.

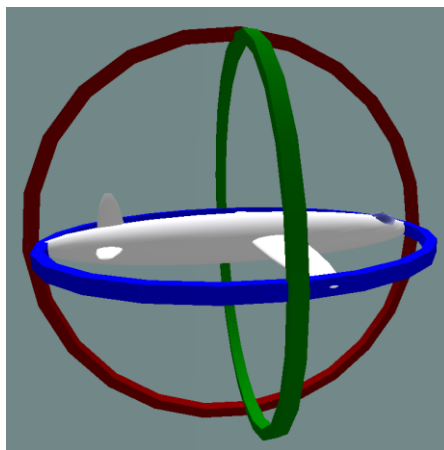


Figura 22 - Situação normal: os três eixos são independentes [36].

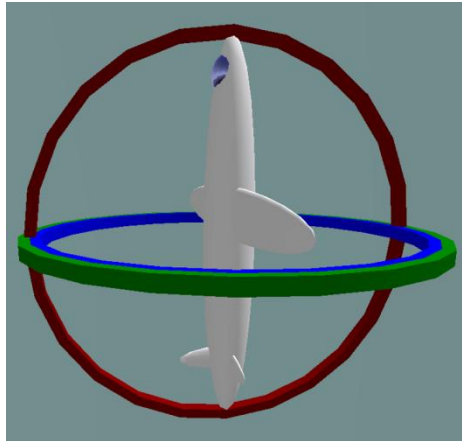


Figura 23 - *Gimbal Lock*: dois dos três eixos estão no mesmo plano, perdendo-se um grau de liberdade [36].

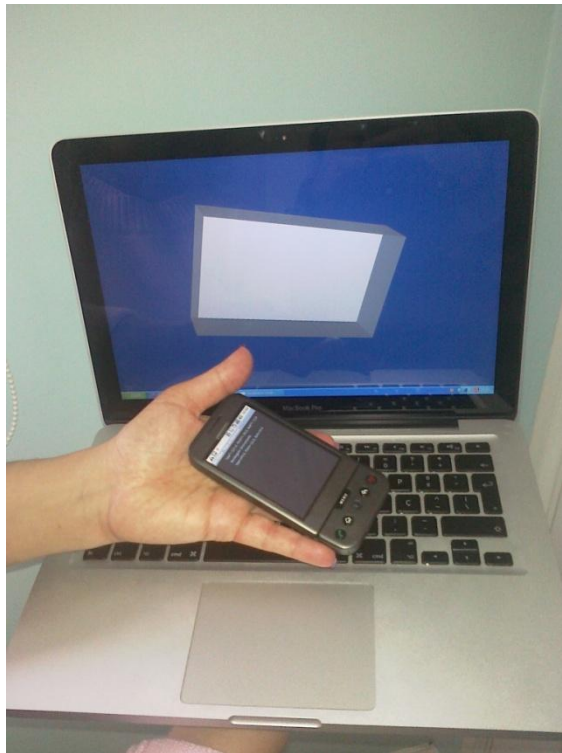


Figura 24 - Sincronização entre o dispositivo móvel e o sólido.

4.2.2. Jogo

O protótipo desenvolvido foi baseado no jogo existente no dispositivo *Android* utilizado, *Labyrinth Lite 3D*, e é descrito a seguir.

Utilizando os valores fornecidos pelo acelerómetro do dispositivo móvel,

criou-se uma aplicação que permite o manuseamento de uma bola dentro de uma plataforma, com o objectivo de fazer cair a bola nos buracos negros, evitando os círculos coloridos. Esta aplicação apresenta duas versões: uma usando apenas o acelerómetro, o que obriga a utilizar o dispositivo móvel sempre na mesma posição para haver sincronismo entre o movimento realizado com o telemóvel e o movimento descrito pela bola; a outra versão utiliza a bússola, para além do acelerómetro, o que permite detectar a orientação do telemóvel, e corrigir o movimento descrito pela bola conforme a posição do dispositivo móvel no espaço. No anexo E está disponível o código implementado para a correcção do movimento da bola. Na figura 25 podemos visualizar um aspecto do resultado final do jogo desenvolvido.

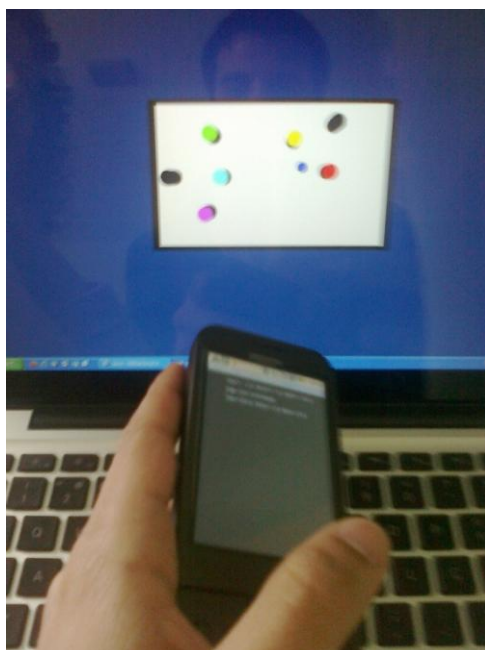


Figura 25 – Aspecto do jogo desenvolvido.

4.3. DetiGuide

O *DetiGuide* é uma aplicação baseada no site do Departamento de Electrónica, Telecomunicações e Informática (DETI) da Universidade de Aveiro. Foi desenvolvida para permitir o uso de um telemóvel *Android* para interagir com um ecrã de grandes dimensões, colocado na entrada do Departamento, de modo a obter informações sobre as actividades e as pessoas do DETI.

A plataforma foi desenvolvida no *Visual Studio 2008* num esquema de *forms*, utilizando a linguagem C#, devido à celeridade de desenvolvimento que esta opção iria permitir. No anexo F é explicada a implementação do *DetiGuide*, sendo apresentados algumas linhas de código.

As formas de interacção desenvolvidas foram escolhidas com base nas necessidades inerentes à consulta de informação num computador. O protótipo desenvolvido permite duas formas de interacção: navegação através de páginas seleccionando a opção da seta da esquerda/direita/cima/baixo (Figura 28) e selecção (Figura 29), similar à interacção do rato, de *links* dentro de páginas (por exemplo seleccionar um nome na página da lista de contactos). Um menu e dois botões virtuais são fornecidos no telemóvel para comutar entre os dois modos de interacção e para duas das operações principais: Selecção e a *homepage* do *DetiGuide*. As Figuras 26 e 27 mostram os menus que integram a aplicação *Android* que controla o *DetiGuide*.

O *DetiGuide* pode ser configurado para usar tanto o acelerómetro/bússola digital ou o ecrã sensível ao toque como forma de interacção no sistema de acordo com os seguintes movimentos:

- Acelerómetro/Bússola Digital: a navegação é feita através de movimentos efectuados com o telemóvel para cima, baixo, esquerda e direita para mudar para a página correspondente. A selecção é feita usando a metáfora de uma bola em cima de um plano que se move de acordo com a gravidade e com os movimentos do telemóvel: pequenos movimentos do dispositivo móvel resultam em movimentos do cursor no ecrã;

- Ecrã sensível ao toque: para navegar através de páginas, os utilizadores arrastam o dedo sobre o ecrã na direcção da página correspondente. A selecção é feita usando o ecrã do telemóvel para mover o cursor de uma maneira semelhante ao *touchpad* de um computador portátil, com a diferença que esta interacção apenas detecta a posição inicial e final do movimento realizado.

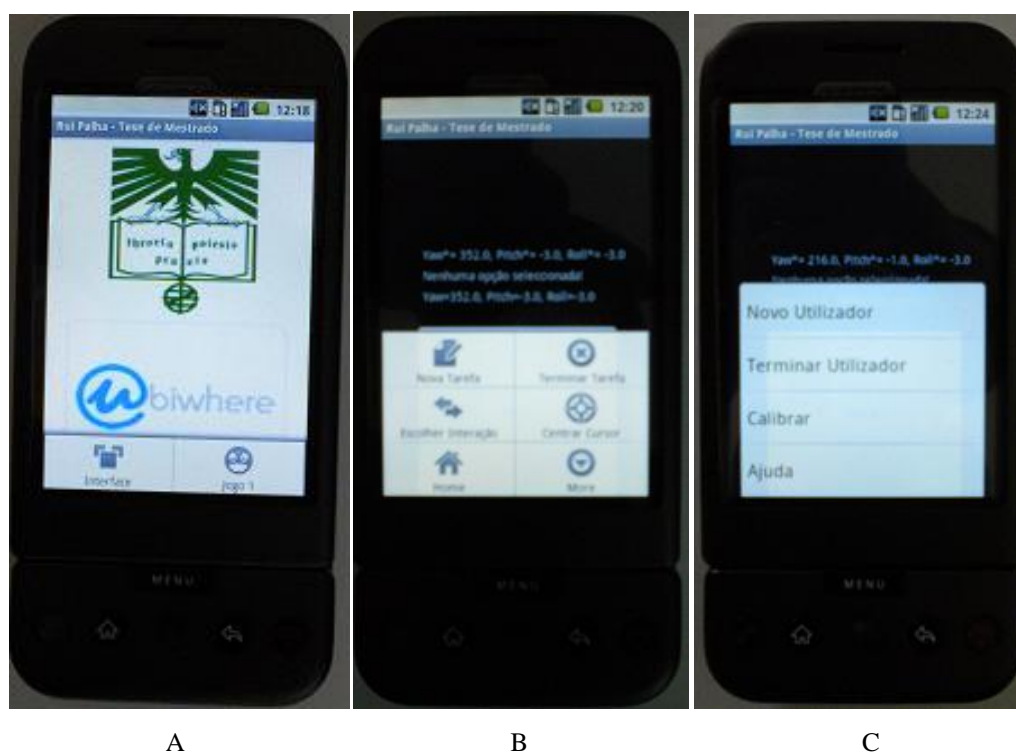


Figura 26 - Imagens da aplicação *Android* que controla o *DetiGuide*: menu inicial (figura A), menu principal (figura B) e menu secundário (figura C).

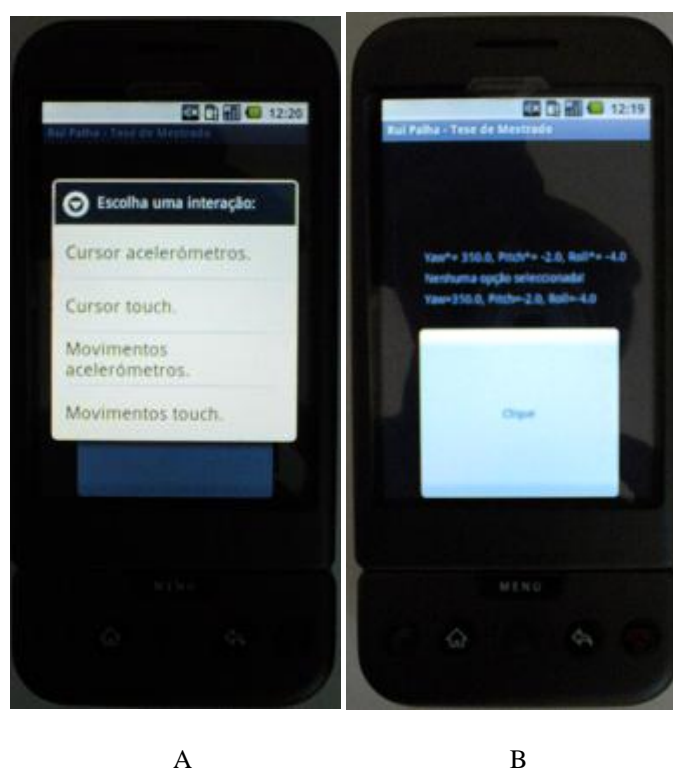


Figura 27 - Imagens da aplicação *Android* que controla o *DetiGuide*: menu de escolha de interações (figura A) e menu de navegação (figura B).



Figura 28 - Exemplo de um utilizador a utilizar o *DetiGuide* em modo de navegação usando o acelerómetro e a bússola digital. Os botões virtuais são visíveis no ecrã do telemóvel.

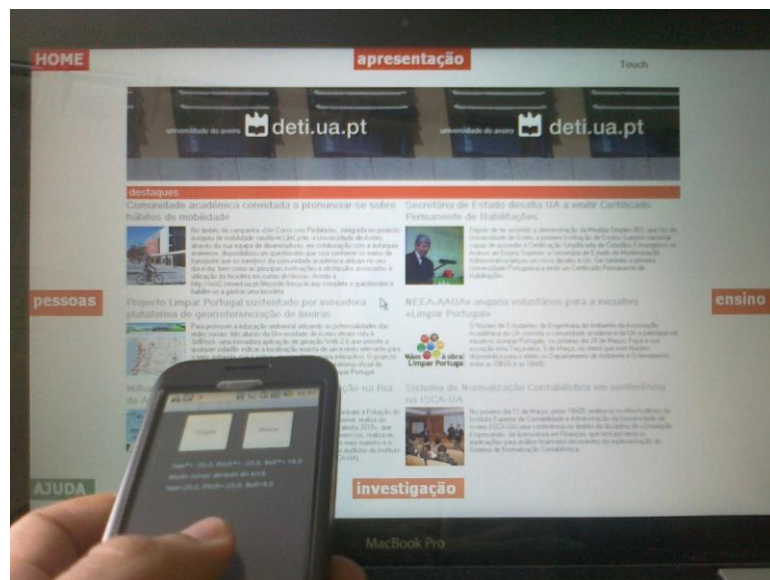


Figura 29 - Exemplo de um utilizador a utilizar o *DetiGuide* em modo de selecção usando o ecrã sensível ao toque.

5. Testes com Utilizadores

Neste capítulo é introduzido o conceito de usabilidade de uma aplicação e apresentados resumidamente os principais métodos de avaliação de interfaces. Descrevem-se as experiências realizadas com utilizadores para testar as diversas formas de interacção com o *DetiGuide*, e são apresentados os resultados.

5.1. Usabilidade

No desenvolvimento de um sistema interactivo é preciso não esquecer os utilizadores a quem este sistema se destina, bem como as tarefas que estes vão desempenhar com a ajuda do sistema. Apesar disto, muitos sistemas têm sido desenvolvidos só a pensar na funcionalidade e não no utilizador. Segundo Jacob Nielsen [37], para uma boa usabilidade é preciso ter em atenção os seguintes aspectos:

- Facilidade de aprendizagem;
- Inexistência de erros;
- Eficiência de utilização;
- Facilidade de lembrar;
- Satisfação.

Segundo *Alan Dix et al* existem regras gerais e regras mais específicas para determinar a usabilidade, os Princípios, as Normas e as Directivas. [38]

- Princípios

Os Princípios são regras gerais e teóricas que se podem aplicar a qualquer tecnologia. São divididas em 3 grupos:

- Facilidade de Aprendizagem

Qualquer utilizador com pouco treino deve ser capaz de utilizar o sistema com um bom desempenho. Além disso deve poder prever qual o resultado de uma dada acção e quais as suas consequências. O sistema deve causar uma primeira boa impressão ao utilizador.

- Flexibilidade

O sistema deve ser o mais ajustável possível aos diferentes utilizadores. Deve ser capaz de suportar mais que uma tarefa e de dar ao utilizador a sensação de que é ele que controla o sistema.

- Robustez

É o feedback que o sistema dá ao utilizador que permite avaliar o estado interno do sistema e corrigir erros cometidos. O tempo de resposta do sistema perante uma acção do utilizador deve ser tão curto quanto possível.

- Normas

Ao contrário dos princípios as normas são regras específicas. Estas normas foram aceites por comunidades internacionais e foram publicadas por instituições como a *International Organization for Standardization (ISO)* e a *British Standards Institution (BSI)*. A parte 11 da norma ISO 9241 [39], chamada por *Ergonomic Requirements for Office Work with Visual Display Terminals*, inclui a *Guidance on usability* que define palavras como, utilizador, tarefa, usabilidade, eficiência e eficácia. A *Guidance on usability* é utilizada como guia, com princípios e técnicas gerais, para a quantificação da usabilidade.

- Directivas

As directivas situam-se entre as normas e os princípios, pois são mais gerais do que as normas, mas mais específicas que os princípios. No entanto é muito difícil concebê-las, pois a usabilidade é um processo composto por diferentes variáveis, logo não existe uma única solução ou regra aplicável; no entanto, estas directivas podem auxiliar na análise e desenvolvimentos alternativos.

5.2. Avaliação de Interfaces

Para melhorar a usabilidade do sistema desde o início do projecto até ao final deve-se avaliar a usabilidade encontrando assim possíveis problemas e corrigi-los [38, 40]. Existem dois tipos de estilos de avaliação:

- No laboratório, o que permite a utilização de material mais sofisticado, assim como fornece um maior controlo sobre a experiência. Contudo, retira naturalidade à experiência, o que poderá condicionar os resultados obtidos.
- No campo, onde os resultados obtidos são mais naturais, apesar de haver menor controlo sobre a experiência, podendo surgir interrupções, o que torna a avaliação mais difícil.

Existem dois tipos de métodos de avaliação [37, 41, 42]:

- Analíticos, sem a participação de utilizadores. São normalmente aplicados na fase de design das aplicações. Dentro dos métodos analíticos, os mais utilizados são:

- Avaliação Heurística

Este método avalia uma interface de utilizador segundo uma lista de princípios. Com esta lista o avaliador é capaz de identificar um problema e classificá-lo segundo a sua gravidade. Jakob Nielsen e Rolf Molich elaboraram uma lista com 10 heurísticas baseadas nos princípios e directivas de usabilidade.

- *Cognitive Walkthrough*

Este tipo de avaliação pode ser usado em qualquer fase do projecto. Este método determina a facilidade de aprendizagem do sistema e, quando se consegue aprender somente explorando o mesmo. Os avaliadores, através da simulação das acções, prevêm problemas de usabilidade, não sendo necessário realizar testes com utilizadores reais.

- Empíricos, com a participação de utilizadores. São frequentemente utilizados na fase de testes das aplicações. Dentro dos métodos empíricos, os mais utilizados são:

- Questionários e Entrevistas

São métodos simples, relativamente baratos e directos. Nos questionários as perguntas são fixas e gerais, utilizadores mais tímidos podem expressar a sua opinião livremente sem haver o confronto cara a cara. As entrevistas são uma técnica mais directa que os questionários e onde a flexibilidade é maior.

- Observação

É o método mais directo, onde são dadas tarefas ao utilizador para este as executar, permitindo medir tempos de execução e observar as suas dificuldades, o que permite obter conclusões sobre a usabilidade do sistema.

- Experiências Controladas

Neste método é necessário escolher uma hipótese a testar. O número de condições experimentais é ponderado por variáveis controladas. É necessário ter atenção a alguns factores: escolha dos participantes, número de participantes, variáveis de teste e hipótese testada.

As experiências manipulam e medem variáveis sobre condições controladas, que permitam testar a hipótese. Existem dois tipos de variáveis, as variáveis independentes que podem ser manipuladas e as dependentes que podem ser medidas. As variáveis independentes são elementos das experiências que podem ser manipulados para produzirem diferentes condições de comparação. Por outro lado, as variáveis dependentes podem ser medidas, o seu valor está dependente das mudanças provocadas pelas variáveis independentes. Estas variáveis formulam a hipótese.

O primeiro passo a ser tomado para uma experiência bem sucedida é formular a hipótese, de seguida é necessário escolher as variáveis dependentes e independentes.

O passo seguinte é decidir pelo método *within groups* (quando o mesmo utilizador testa a experiência em condições diferentes) ou pelo método *between groups* (quando parte dos utilizadores participam numa condição e os restantes utilizadores noutra condição).

Por último efectuem-se os testes e analisam-se os resultados obtidos.

5.3. Experiências Realizadas

De modo a validar as opções feitas e avaliar as interacções disponíveis no *DetiGuide*, foram realizados testes com utilizadores. Foi então elaborada uma lista de tarefas para observar a forma como o utilizador reagia a cada uma delas. As tarefas foram escolhidas com base na importância que desempenham no contexto do *DetiGuide*.

A lista de tarefas para os utilizadores realizarem era composta por 4 perguntas, cujas respostas podiam ser encontradas pesquisando o *DetiGuide*. As tarefas eram as

seguintes (no Anexo G encontra-se uma lista mais detalhada das tarefas):

1. Em que ano foi fundado o Departamento?
2. Qual o telefone de contacto de um determinado docente?
3. Em que ano foi atribuído ao Instituto um dado estatuto?
4. Qual o código no Portal Académico de uma dada disciplina?

Duas das tarefas implicavam que o utilizador usasse apenas as interacções com o acelerómetro/bússola digital, enquanto nas outras duas tarefas só poderia utilizar as interacções com o ecrã sensível ao toque. Metade dos utilizadores utilizaram as duas formas de interacção (*touch* e acelerómetros) numa dada ordem e a outra metade em ordem inversa para se poderem compensar e estudar eventuais possíveis efeitos de aprendizagem.

Enquanto na primeira e na terceira tarefa apenas era necessário a utilização do modo de navegação, a segunda e a quarta tarefa exigiam o uso do modo de navegação e de selecção. Na segunda tarefa era ainda necessária a utilização do *scroll* dado a lista de docentes não ser toda visível numa só página.

Os utilizadores não tiveram a possibilidade de experimentarem o sistema antes de efectuarem os testes, sendo apenas demonstrado aos utilizadores os modos de interacção disponíveis. Esta decisão prendeu-se com o facto de tentar averiguar se existiu uma aprendizagem na utilização dos modos de interacção. No final dos testes, foi pedido aos utilizadores que respondessem a um questionário recolhendo informações sobre a experiência deles com este tipo de tecnologias e a sua opinião quanto a alguns aspectos do *DetiGuide* (ver anexo H).

Caracterização da Amostra

Os utilizadores escolhidos para a realização dos testes foram alunos da Universidade de Aveiro dada a facilidade de acesso aos mesmos. Este facto condicionou os resultados obtidos, já que todos os utilizadores são bastantes familiarizados com a tecnologia em geral. Foram realizados testes com um total de 51 utilizadores em 2 sessões, 26 na 1ª sessão e 25 na 2ª Sessão, numa faixa etária entre os 19 e 46 anos, e com uma percentagem de 96% dos utilizadores do sexo masculino e 4% do sexo feminino. Esta amostra, embora não seja representativa do público em geral, é muito representativa dos possíveis utilizadores do *DetiGuide*.

Caracterização do Ambiente de Trabalho

Os testes realizaram-se na entrada do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro no dia 20 de Abril e 24 de Maio de 2010. Foi utilizado um dos ecrãs disponíveis no local, um computador pessoal e um telemóvel *HTC Dream*. Foi ainda utilizado o *Visual Studio 2008* para correr o *DetiGuide*.

No início dos testes, foi feita uma apresentação, de cerca de 5 minutos, numa sala com os alunos que iriam realizar os testes. Apenas um utilizador se encontrava junto ao ecrã para a realização do teste, sendo-lhe demonstrado o funcionamento do *DetiGuide* no início do teste.

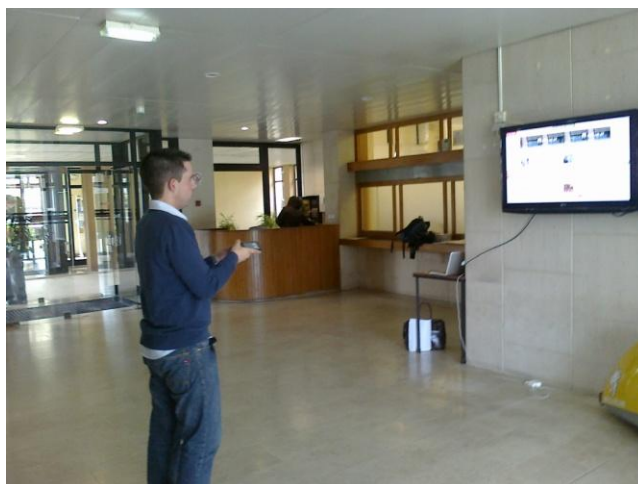


Figura 30 - Utilizadores usando o *DetiGuide*.



Figura 31 - Utilizadores usando o *DetiGuide*.

5.4. Resultados Obtidos

De acordo com os dados obtidos através dos testes com utilizadores, foram elaborados gráficos e obtidas conclusões que são apresentadas a seguir. No anexo I encontram-se os dados obtidos na 1ª sessão de testes com os utilizadores e no anexo J encontram-se os dados obtidos na 2ª sessão de testes com os utilizadores.

Alguns utilizadores na primeira sessão de testes queixaram-se de que a selecção de itens no ecrã era pouco confortável devido ao tamanho reduzido do botão de selecção. Devido a este facto, o tamanho do botão de selecção foi aumentado da primeira para a segunda sessão de testes. Essa alteração teve influência nos resultados da segunda sessão de testes, comprovada pela diminuição do tempo das tarefas, em especial aquelas em que era necessário utilizar o botão de selecção (Tarefas 2 e 4), e também pela opinião dos utilizadores que atribuíram à Tarefa 4, uma das afectadas pelo tamanho do botão de selecção, um grau de dificuldade menor.

O tempo total e o grau de dificuldade das tarefas que foram afectadas na segunda sessão de testes por esta alteração, serão analisados separadamente dos da primeira sessão de testes, enquanto os que não foram afectados serão analisados em conjunto.

Análise do Tempo das Tarefas

O tempo total das tarefas refere-se ao tempo desde que o utilizador saiu da página inicial do *DetiGuide* até ao instante em que chegou à página que continha a informação necessária para responder à tarefa.

Com base nas Figura 32 e 33, podemos verificar que as Tarefas 2 e 4 apresentaram um maior grau de dificuldade em relação às Tarefas 1 e 3, visto que os utilizadores demoraram mais tempo para a execução destas tarefas, tal como era esperado. A Tarefa 2 foi a mais demorada pois para além de ser necessária a utilização do modo de navegação e selecção como na Tarefa 4, era também necessário o uso do *scroll*.

Podemos também afirmar que os utilizadores demoraram mais tempo a executar as tarefas com o acelerómetro/bússola digital do que com o ecrã sensível ao toque, com excepção da Tarefa 4, o que sugere que os utilizadores tiveram maiores dificuldades na

utilização do acelerómetro/bússola digital.

Da primeira para a segunda sessão de testes, verifica-se uma ligeira diminuição do tempo médio total das tarefas, sobretudo na Tarefa 2.

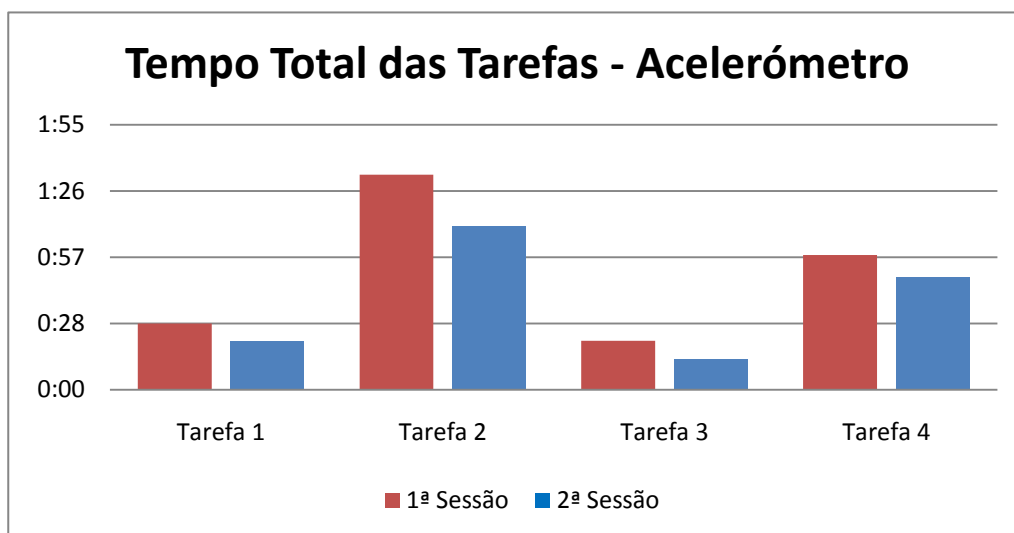


Figura 32 - Tempo médio total das tarefas nos testes com utilizadores utilizando a interacção através do acelerómetro/bússola digital.

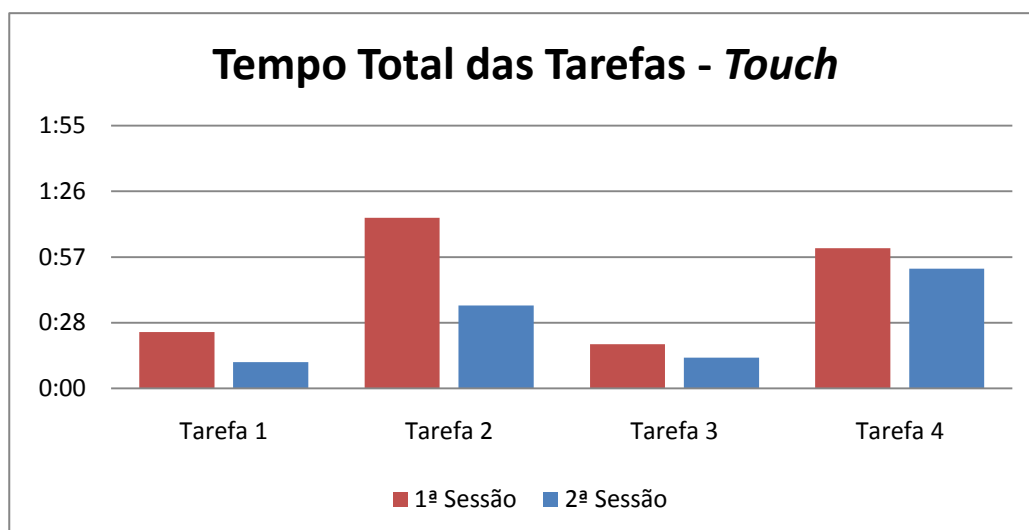


Figura 33 - Tempo médio total das tarefas nos testes com utilizadores utilizando a interacção através do ecrã sensível ao toque.

Grau de Dificuldade das Tarefas

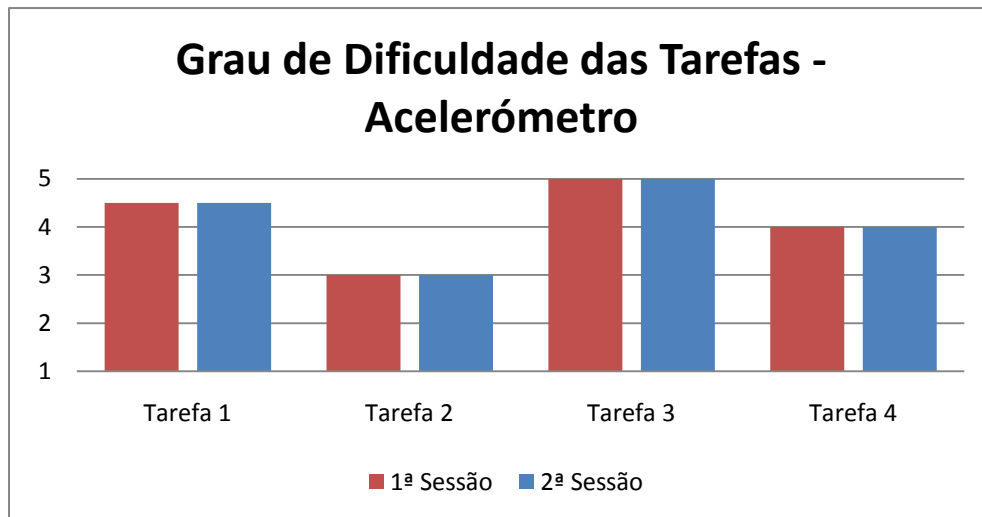


Figura 34 - Grau de dificuldade das tarefas nos testes com utilizadores utilizando a interação através do acelerómetro/bússola digital.

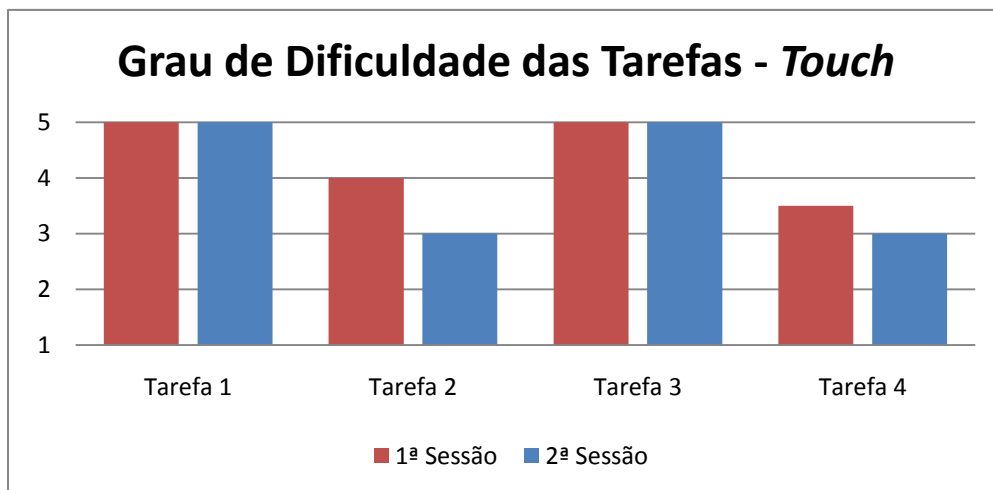


Figura 35 - Grau de dificuldade das tarefas nos testes com utilizadores utilizando a interação através do ecrã sensível ao toque.

Para o registo do grau de dificuldade das tarefas observado pelo observador e indicado pelos utilizadores foram usadas escalas de *Lickert* de 5 níveis, com valores entre 1 e 5 (1 Muito Difícil e 5 Muito Fácil). Com base nas Figuras 34 e 35, podemos verificar que as Tarefas 2 e 4 foram consideradas como tendo um grau de dificuldade maior do que as Tarefas 1 e 3, como era esperado. Esta conclusão deve-se ao facto de as Tarefas 2 e 4

exigirem a utilização de duas formas de interacção no *DetiGuide*, navegação e selecção, este último mais complexo de controlar, enquanto as Tarefas 1 e 3 apenas necessitavam de uma forma de interacção, a navegação.

Os utilizadores acharam a primeira tarefa mais fácil através do ecrã sensível ao toque e na tarefa 3 o nível de dificuldade experimentado pelos utilizadores foi o mesmo em ambas as interacções. Relativamente à tarefa 2 os utilizadores acharam que na primeira sessão de testes o nível de dificuldade foi maior através da interacção com o acelerómetro/bússola digital, e na segunda sessão de testes o nível de dificuldade experimentado pelos utilizadores foi o mesmo. Na tarefa 4 os utilizadores acharam que foi mais fácil a interacção através do acelerómetro/bússola digital.

Número de Erros nas Tarefas

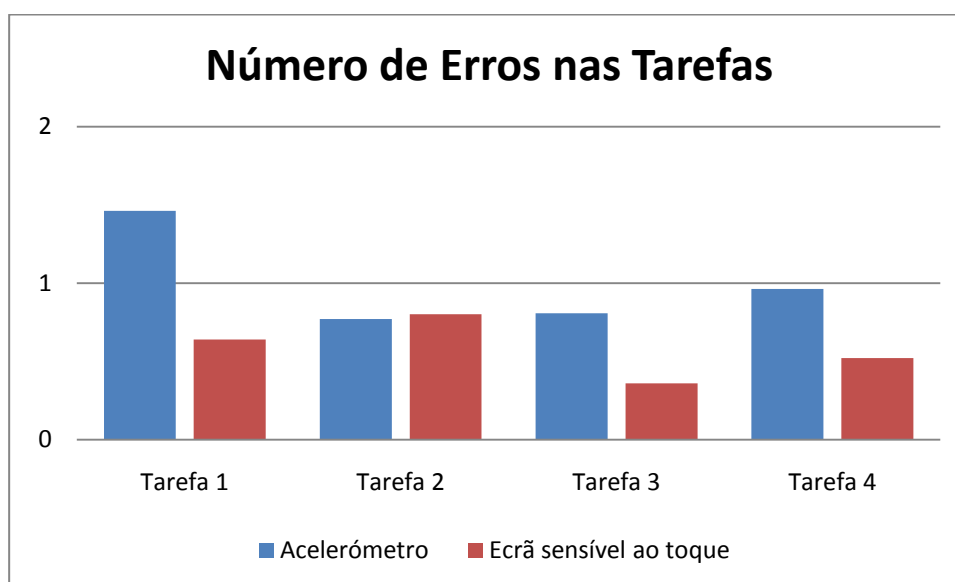


Figura 36 - Número de Erros nas Tarefas dos Testes com Utilizadores.

O número de erros nas tarefas foi contabilizado partindo do princípio que um erro consiste na visualização de um ecrã desnecessário para a realização da tarefa. Com base na Figura 36, podemos verificar que os utilizadores realizaram um maior número de erros quando utilizaram o acelerómetro/bússola digital do que com o ecrã sensível ao toque, com excepção da Tarefa 2, o que reforça a ideia de que os utilizadores tiveram maiores dificuldades utilizando o acelerómetro/bússola digital. Podemos constatar também que foi

efectuado um número menor de erros na realização da Tarefa 2 usando o acelerómetro/bússola digital do que na Tarefa 1, apesar da Tarefa 2 possuir um grau mais elevado de dificuldade, o que sugere que houve uma rápida aprendizagem por parte dos utilizadores na utilização da interacção através do acelerómetro/bússola digital. Este facto poderá ser justificado por esta tarefa ser bastante intuitiva.

Opinião dos Utilizadores

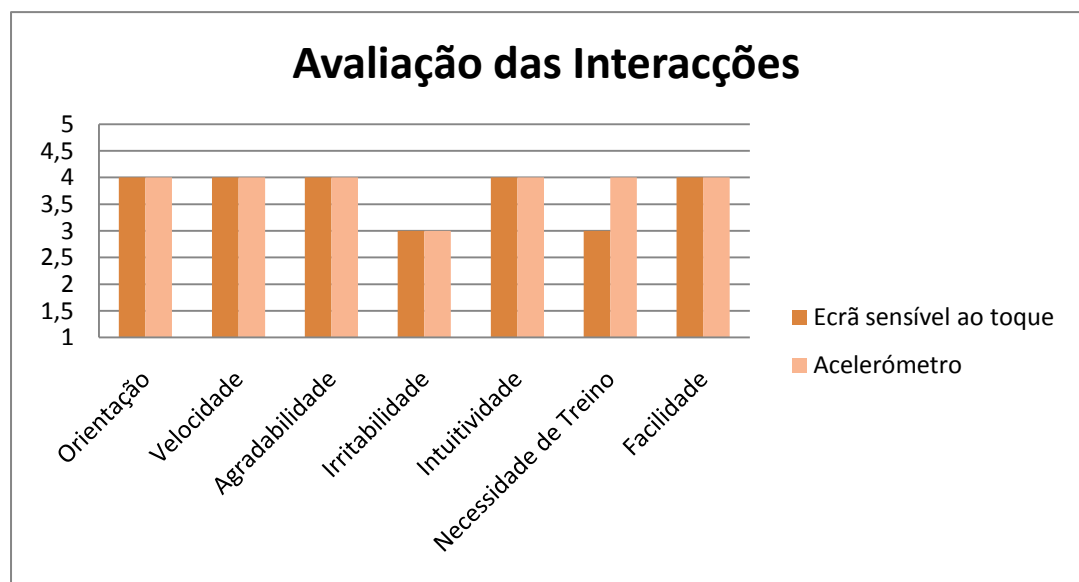


Figura 37 - Resultados dos Questionários dos Utilizadores após a realização das tarefas.

As duas formas de interacção foram avaliadas por cada utilizador no final dos testes através de um questionário, onde foram manifestadas a concordância ou discordância em relação às afirmações apresentadas, onde 1 representa Discordância Total e 5 Concordância Total.

Com base na Figura 37, podemos concluir que os utilizadores acharam que as duas formas de interacção disponíveis eram bastante usáveis e equilibrados. Destaque-se o facto de os utilizadores acreditarem que a interacção através do acelerómetro/bússola digital necessitar de mais treino do que com o ecrã sensível ao toque, o que é normal visto que a utilização do acelerómetro/bússola digital é ainda pouco explorada no mercado, enquanto o ecrã sensível ao toque é uma tecnologia bastante divulgada no mercado de dispositivos móveis.

Satisfação e Preferência dos Utilizadores

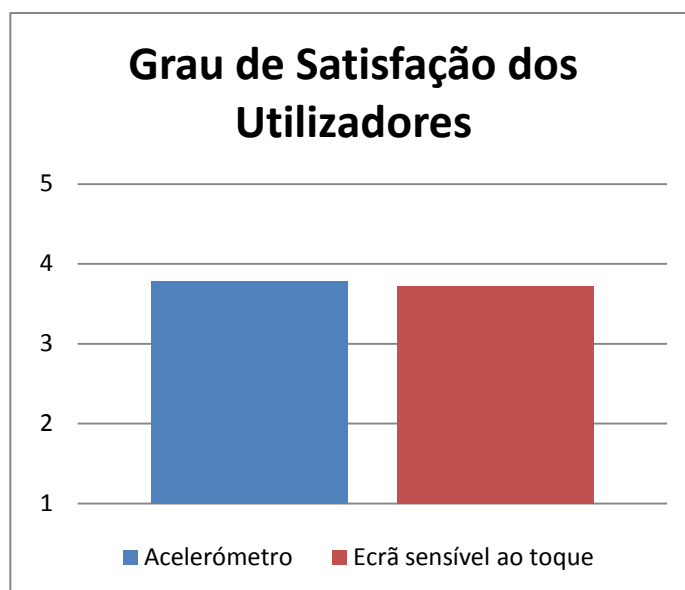


Figura 38 - Grau de Satisfação dos Utilizadores.

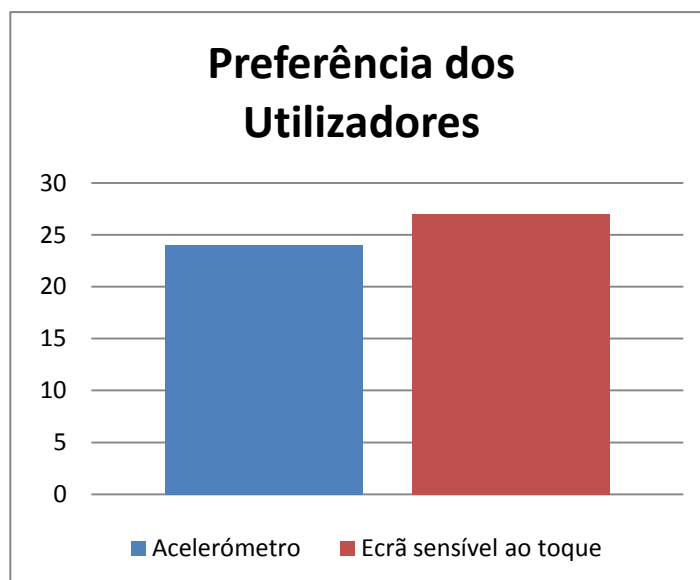


Figura 39 - Preferência dos Utilizadores.

O Grau de Satisfação dos Utilizadores na utilização de cada uma das formas de interacção foi indicado com valores entre 1 e 5 (1 Nada Satisfeito e 5 Muito Satisfeito).

Observando a Figura 38 e 39, verifica-se que os utilizadores consideraram as duas formas de interacção disponíveis equilibradas, assim como que ambas apresentavam uma

boa usabilidade. Apesar de equilibradas, a interacção através do ecrã sensível ao toque foi a preferida dos utilizadores, com 27 utilizadores a preferirem esta interacção contra 24 que preferiram o acelerómetro/bússola digital. Esta conclusão justifica-se já que os utilizadores estão mais experimentados com esta forma de interacção, visto existirem bastantes dispositivos móveis com esta tecnologia no mercado, ao contrário da tecnologia com acelerómetro/bússola digital, ainda pouco explorada. Outros factos que apoiam esta escolha são o maior número de erros e o maior tempo necessário para realizar as tarefas com o acelerómetro/bússola digital.

6. Conclusões e Trabalho Futuro

Neste capítulo faz-se uma apreciação global ao trabalho realizado tendo em consideração os resultados obtidos, são apresentadas também algumas propostas para trabalho futuro.

6.1. Análise Global

As aplicações desenvolvidas ao longo deste projecto foram o controlo de um sólido, um jogo e o *DetiGuide*. Todas estas aplicações são controladas através dos movimentos efectuados com o dispositivo móvel *Android*. O controlo do sólido permitiu aprender a usar e a aceder aos diversos sensores do dispositivo móvel utilizado, e também testar a arquitectura de comunicação implementada, especificamente a velocidade e a fiabilidade da comunicação. Numa fase posterior, a criação do jogo possuiu como objectivo testar o potencial do acelerómetro para interacção. Por último, o propósito do *DetiGuide* foi testar diferentes formas de interacção possíveis usando um dispositivo móvel.

Desenvolvimento

A sincronização do dispositivo móvel com o sólido e o desenvolvimento do jogo foram o primeiro passo dado no sentido de utilizar as capacidades dos sensores do telemóvel *Android* assim como da comunicação entre o telemóvel e o ecrã. A principal

conclusão a reter é que a escolha do protocolo usado para estabelecer a comunicação entre o telemóvel *Android* e o servidor do computador tem de ser cuidadosa devido ao facto de poder comprometer o desempenho do sistema.

Numa segunda fase, o desenvolvimento da plataforma *DetiGuide* permitiu criar um sistema similar ao que era pensado implementar em museus ou locais públicos da cidade. Contudo, existem algumas questões sobre a usabilidade do sistema que merecem alguma reflexão, assim como certas funcionalidades que ainda é necessário implementar para tornar o sistema comercializável, de acordo com os resultados obtidos nos testes com utilizadores.

Resultados com os Utilizadores

No final dos testes ao *DetiGuide*, foram retiradas algumas conclusões, fundamentadas pela opinião dos utilizadores sobre a sua utilização e também pela análise dos resultados obtidos. As conclusões foram as seguintes:

- A interacção através do ecrã sensível ao toque causou alguma irritação junto dos utilizadores pelo facto de este sistema não ter um comportamento similar ao de um *touchpad* de um computador portátil. Devido a limitações do dispositivo, não foi possível alterar esta característica do sistema;
- O movimento efectuado com o telemóvel durante a interacção através do acelerómetro/bússola digital foi considerado pouco intuitivo por parte de alguns utilizadores, já que é necessário deslocar o telemóvel para baixo para seleccionar uma opção na parte de cima do ecrã e vice-versa. Contudo, analisando os resultados dos testes, a maioria, 63%, concordou com a atribuição efectuada aos movimentos;
- Na 1ª sessão de testes, quando foi utilizada a opção de selecção com o acelerómetro/bússola digital, o controlo do cursor foi considerado pouco confortável por parte de alguns utilizadores, devido ao botão para accionar o clique do rato ser considerado pequeno. Na 2ª sessão de testes, o botão de clique do rato foi aumentado, passando a ocupar cerca de metade do ecrã do telemóvel. Apesar do tempo total das tarefas ter diminuído nas tarefas afectadas pela utilização daquele botão, o espaço para manipulação da

interacção do ecrã sensível ao toque diminuiu consideravelmente, o que poderá afectar alguns parâmetros da navegação, como torná-la mais irritante e menos agradável, o que não se verificou;

- Na interacção através do ecrã sensível ao toque, os utilizadores acharam que o cursor deveria permitir maior precisão do movimento. No entanto, devido ao espaço disponível no ecrã para efectuar esta interacção, principalmente depois de aumentar o botão de clique do rato, o aumento de precisão do rato aumentaria consideravelmente o número de movimentos necessários que o utilizador teria de efectuar para conseguir atingir o item pretendido, o que causaria maior irritação junto do utilizador, assim como aumentaria tempo total das tarefas;
- Apesar de na primeira sessão de testes ter ficado a sensação que teria existido uma aprendizagem por parte dos utilizadores que iniciaram as tarefas com a interacção do acelerómetro/bússola digital, já que verificou-se uma diminuição gradual do número de erros efectuados, a segunda sessão mostrou que tal não se confirmou, tendo sido consequência de uma amostragem reduzida.

Conclusões Gerais

Na medida em que o trabalho desenvolvido foi exploratório, algumas opções foram tomadas com base no tempo disponível para a sua implementação. O tempo dispendido para conseguir a sincronização entre o dispositivo móvel e o sólido foi maior que o esperado, devido às várias tentativas efectuadas para realizar esta operação, o que acabou por causar uma redução do tempo disponível para desenvolver as restantes aplicações.

No final dos trabalhos realizados, podemos obter algumas conclusões que podem ser úteis para desenvolver trabalho futuro. Ao contrário do que se esperava inicialmente, a interacção através do acelerómetro/bússola digital superou bastante as expectativas, mostrando ser uma séria opção para interacção a utilizar em futuras aplicações. A escolha das formas de interacção tem que ser muito bem pensada, de acordo com as características

das aplicações nas quais as interacções vão ser integradas. Terá que existir uma relação de compromisso entre os vários parâmetros a considerar, como o tamanho dos botões da interface e a adequação da sensibilidade dos sensores.

Apesar de necessitar de bastante tempo de aprendizagem, a plataforma *Android* demonstrou o seu potencial durante a realização do trabalho desenvolvido, mostrando ser uma opção para dispositivo de entrada de um sistema multimédia interactivo.

6.2. Trabalho Futuro

Apesar das várias etapas superadas com sucesso, existem alguns aspectos que poderão ser melhorados para aumentar a qualidade do software desenvolvido:

- Incluir na opção de selecção da interacção através do acelerómetro/bússola digital um movimento “inteligente” na vizinhança de *links*, que consiste em colocar o cursor a descrever uma trajectória automática na direcção do objecto mais próximo do movimento efectuado com o telemóvel;
- Realizar uma outra experiência com utilizadores, testando a precisão do acelerómetro e do ecrã sensível ao toque, de modo a tentar encontrar soluções para melhorar o controlo do cursor do *DetiGuide* através dos sensores do telemóvel. Para isso, poder-se-ia adaptar o jogo desenvolvido durante este projecto, de modo a que o movimento da bola também pudesse ser controlado com o ecrã sensível ao toque, e avaliar o desempenho do jogo comparando as duas formas de interacção;
- Tornar o *DetiGuide* uma plataforma portátil, de modo a facilitar a introdução da informação mostrada na aplicação, assim como torná-la acessível de qualquer local. Espera-se que novas versões sejam desenvolvidas utilizando plataformas Web, o que aumentaria significativamente a portabilidade do sistema, podendo esta ser acedida de qualquer local com ligação à Internet e a partir de qualquer sistema operativo;
- Concretizar a possibilidade do dispositivo móvel ser detectado automaticamente quando este se encontrar próximo de um local onde esteja disponível a plataforma similar ao *DetiGuide*. Para esta situação, e utilizando a última versão do sistema operativo do *Android* disponível aquando da escrita desta dissertação (versão 2.1), o telemóvel poderá ser visível através de *Bluetooth* para o servidor da plataforma, que informa o portador do

dispositivo móvel que pode descarregar a aplicação para o seu telemóvel de modo a interagir com o ecrã;

- De modo a aumentar a versatilidade da aplicação desenvolvida para *Android*, para utilizar a aplicação num local público, será necessário adaptar o software desenvolvido para as várias plataformas móveis que existem no mercado, podendo o sistema operativo do telemóvel ser detectado através de *Bluetooth*, aquando da transferência da aplicação para o telemóvel.

Anexo A – Código implementado em Java para um cliente – servidor TCP/IP

Solução para cliente TCP/IP:

```
import org.apache.http.client.ResponseHandler;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.BasicResponseHandler;
import org.apache.http.impl.client.DefaultHttpClient;

public void TCPClient() {

    DefaultHttpClient client = new DefaultHttpClient();
    HttpGet method = new HttpGet();
    ResponseHandler<String> responseHandler = new BasicResponseHandler();

    method = new HttpGet("http://172.16.44.41:8082?" + "teste");

    try {
        client.execute(method, responseHandler);
    }
    catch (IOException e) {
        e.printStackTrace();
    }
}
```

Solução para servidor TCP/IP:

```
import com.sun.net.httpserver.HttpServer;
import java.net.InetSocketAddress;
import java.util.concurrent.Executors;

public class TCPServer{

    public static void main(String[] args){
        try{
            HttpServer server = HttpServer.create(new
InetSocketAddress("192.168.0.150", 8082), 0);
            server.setExecutor(Executors.newCachedThreadPool());
            server.start();
            server.createContext("/", new TratamentoDados());
            System.out.println("Server Listening");
        }
        catch(IOException e){
            System.err.println("HttpServer Error!");
            e.printStackTrace();
        }
    }
}

import com.sun.net.httpserver.HttpExchange;
import com.sun.net.httpserver.HttpHandler;

public class TratamentoDados implements HttpHandler {

    public void handle(HttpExchange exchange) throws IOException {

        final String dados = exchange.getRequestURI().toString();
        exchange.sendResponseHeaders(200, "Mensagem Recebida!".length());
        OutputStream os = exchange.getResponseBody();
        os.write("Mensagem Recebida!".getBytes());
        os.close();
    }
}
```

Anexo B – Código implementado em Java para um cliente – servidor UDP

Solução para cliente UDP:

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;

public void UDPClient() {

    String sentence = "teste";
    byte[] sendData = sentence.getBytes();

    try {
        clientSocket = new DatagramSocket();
        InetAddress IPAddress;

        try {
            IPAddress = InetAddress.getByName("172.16.10.61");
            DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, IPAddress, 8082);
            clientSocket.send(sendPacket);
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    }
    catch (SocketException e) {
        e.printStackTrace();
    }
}
```

Solução para servidor UDP:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;

public void UDPServer(){
    String data = new String();
    DatagramSocket serverSocket = null;

    try {
        serverSocket = new DatagramSocket(8082);
        byte[] receiveData = new byte[220];

        while(true)
        {
            String aux;
            DatagramPacket receivePacket = new
DatagramPacket(receiveData, receiveData.length);
            serverSocket.receive(receivePacket);
            aux = new String( receivePacket.getData()).trim();
            String[] lista = aux.split("\\\\+");
            data = lista[44];
        }
    }
    catch (IOException e) {
        e.printStackTrace();
    }
}
```

Anexo C - Instalação VTK em Windows com suporte Java

Este documento foi escrito de modo a tentar facilitar para outros utilizadores a instalação, longa e frustrante, das bibliotecas VTK em Windows com suporte Java. A data da criação deste documento é 07 de Maio de 2010. Esta é apenas a minha experiência, portanto não posso garantir que todas as observações aqui apresentadas estão correctas, mas esta configuração e estes passos resultaram no funcionamento correcto das bibliotecas VTK em Java no meu computador.

Na minha configuração, a versão VTK instalada foi a 5.0.4, no *Windows XP Professional Service Pack 3*, para usar com a versão do *Java Development Kit 1.6.0_17* suportada pelo *Eclipse SDK 3.5.1*.

1. Downloads necessários e instalação de software

1.1. Download da *source* VTK

A *source* VTK pode ser transferida do site oficial do VTK, <http://www.vtk.org/>. Após entrar no site, seleccionar o separador *Resources* e seguidamente *Download*. Fazer download da última versão disponível (no meu caso foi *vtk-5.0.4.zip*). Se quiser usar o VTK com suporte Java tem que fazer o download da *source*, e não do *Windows Installer*. A opção de fazer o wrapping em Java é possível durante a compilação manual de VTK, mas com o *Windows Installer* não é possível.

1.2. Download do CMake e instalação

Para compilar VTK utilizei o software CMake. CMake é usado para controlar o processo de compilação de software usando uma plataforma simples e um compilador de arquivos de configuração independentes. O software CMake pode ser transferido do site oficial do CMake, <http://www.cmake.org/>. Após entrar no site, seleccionar o separador *Resources* e seguidamente *Download*. Para este caso já existe instalador para Windows. Fazer download da última versão disponível (no meu caso foi *cmake-2.8.1-win32-x86.exe*).

1.3. Instalação do compilador C++

Eu usei o compilador do Microsoft *Visual Studio 2008*.

1.4. Download do Java SDK e instalação

O download pode ser feito do site oficial da Sun, <http://java.sun.com/>. Deverão transferir o *Java Development Kit (JDK)* e não o *Java Runtime Environment (JRE)*! Após entrar no site, seleccionar o separador Downloads, e seguidamente Java SE. Transferir a última versão disponível (no meu caso foi a JDK 1.6.0_17).

1.5. Download do Eclipse e instalação

O download pode ser feito através do site oficial do Eclipse, <http://www.eclipse.org/>. Eu transferi a versão 3.5.1, a mais recente disponível na altura. Após entrar no site, seleccionar o separador Downloads, e escolher a opção *Eclipse IDE for Java Developers*.

2. Compilar a source VTK com o CMake

Nesta fase já temos o compilador de C++, o CMake e o Java SDK instalado, assim como a *source* VTK disponível. Já estão reunidas todas as condições para começar a compilação da *source* VTK. Começa-se por executar o CMake, onde irá aparecer uma janela como na figura seguinte.

Primeiro, é preciso especificar o *path* onde a *source* VTK irá ser instalada. No meu caso foi C:/Program Files/vtk. Seguidamente, escolher o *path* onde irão ser instalados os binários. No meu caso foi C:/Program Files/vtk/bin. Ter o cuidado de verificar se as opções BUILDING_SHARED_LIBS, VTK_USE_RENDERING e VTK_WRAP_JAVA estão ativas como na figura seguinte. Seleccionar *Configure* seguido de *Generate* para iniciar o processo, que deverá demorar cerca de 30 minutos até estar concluído.

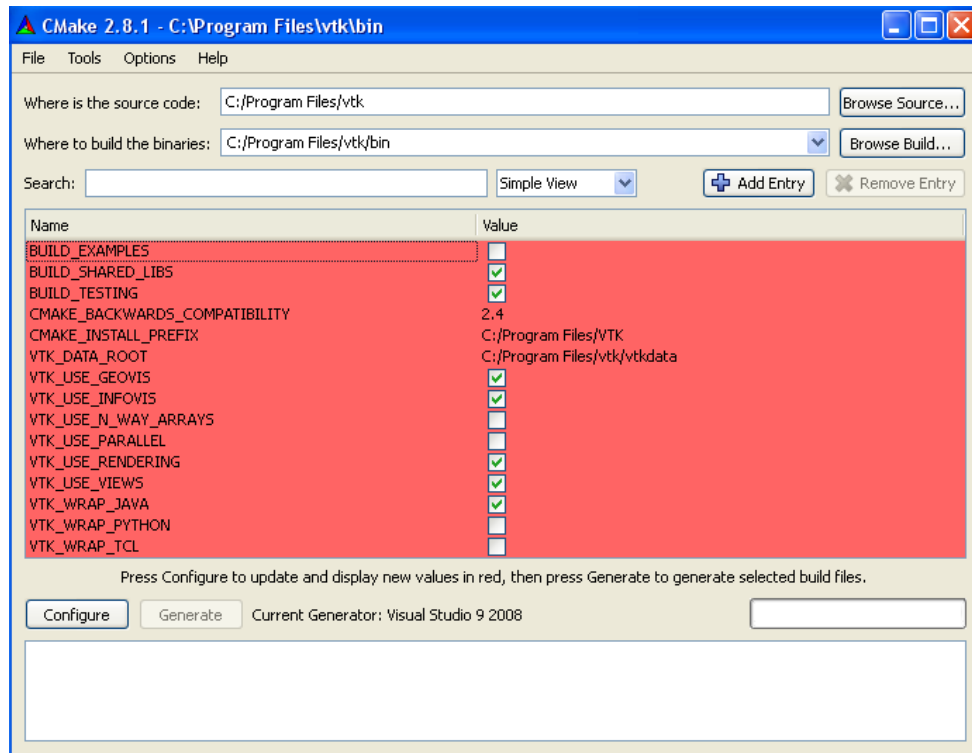


Figura 40 - Janela do CMake.

3. Compilar a configuração no compilador C++ (Microsoft Visual Studio 2008)

Agora, é necessário compilar toda a configuração num compilador C++. Eu utilizei o *Visual Studio 2008*. Para isso, devemos ir à pasta dos binários do VTK. No meu caso, C:/Program Files/vtk/bin. Procurar o projecto chamado ALL_BUILD.vcproj. Fazendo duplo clique neste ficheiro, deverá abrir o compilador C++ previamente escolhido (no meu caso *Visual Studio 2008*) como na figura seguinte. Seleccionar *Solution VTK* (como na imagem) e seleccionar a opção *Build* seguido de *Build Solution*. Este é o processo mais demorado, e o mais normal é serem encontrados erros no fim do processo. No meu caso, os erros estavam relacionados com o *building* dos ficheiros Java. A solução é compilar todas as classes Java com um compilador Java. Eu utilizei o Eclipse.

Uma coisa muito importante a fazer é editar a variável do ambiente PATH. Apesar de ter activado a opção BUILD_SHARED_LIBS, é necessário informar o Windows do local onde se encontram os DLLs. Para aceder ao PATH, ir ao Painel de Controlo,

Performance e Manutenção, Sistema, seleccionar o separador Avançados, e seleccionar Variáveis de Ambiente. Nas Variáveis do Utilizador, editar o PATH, adicionando-lhe o seguinte caminho: C:\Program Files\vtk\bin\bin\Debug.

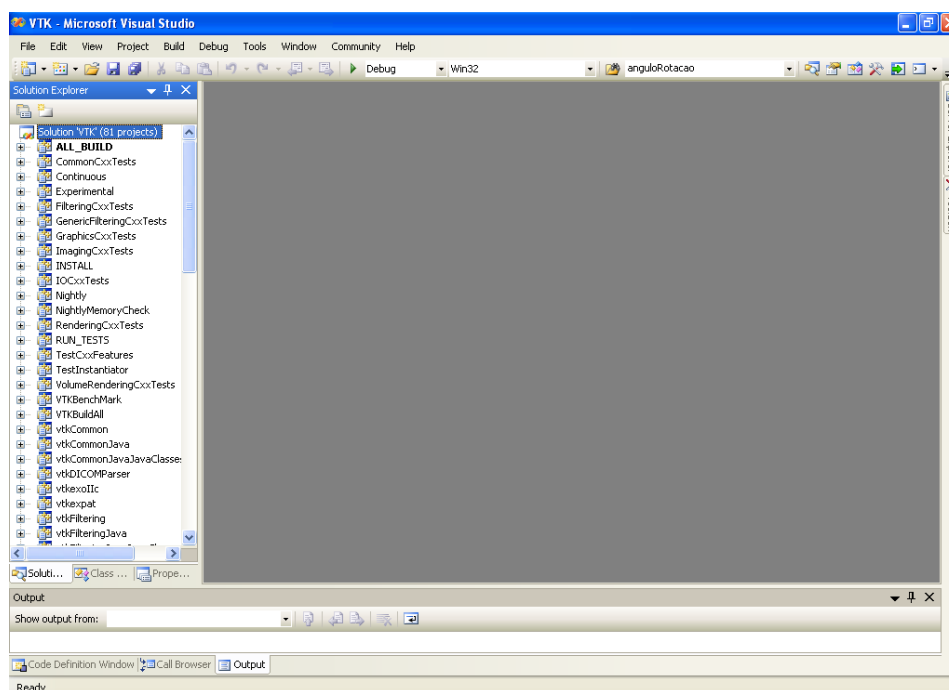


Figura 41 – Janela do Visual Studio 2008.

4. Configuração do ambiente Java no Eclipse

É possível compilar as classes Java de uma maneira muito básica e manual. Primeiro, é necessário localizar as classes. No meu caso, estas encontravam-se em C:\Program Files\vtk\bin\java. Dentro desta pasta encontra-se outra pasta com o nome *vtk* onde se encontram as classes Java (*.java). Agora basta abrir um projecto no Eclipse (File->New->Project->Java Project). O nome do projecto é indiferente. Vou-lhe chamar Novo. Aceda à pasta src do projecto no Eclipse (tipicamente C:\Documents and Settings\User\My Documents\workspace\Novo\src) e copie as classes .java que se encontram em C:\Program Files\vtk\bin\java\vtk para esta pasta. Agora voltando ao Eclipse, faça *Refresh* sobre o projecto New, e aparecerão as classes .java no Eclipse. Automaticamente, irão ser criadas as classes .class associadas aos ficheiros .java. Se obtiver um erro no pacote

VTKJavaWrapped apague-o. Agora, copiar os ficheiros .class que se encontram em C:\Documents and Settings\User\My Documents\workspace\vtk\bin para C:\Program Files\vtk\bin\java\vtk. Agora, nesta pasta encontram-se os ficheiros .java e .class. Compacte os ficheiros .java e .class num só ficheiro .zip, podendo-lhe chamar vtk.zip. Agora inclua este ficheiro .zip em qualquer projecto VTK e deverá estar a funcionar! Para adicionar este ficheiro no Eclipse, clique com o botão direito do rato em cima do projecto, escolha *Properties* seguido de *Java Build Path*. Depois seleccione Add External Jar e escolha o ficheiro vtk.zip. Agora deverá poder compilar e correr uma aplicação VTK em Java sem quaisquer problemas.

Anexo D – Código implementado para aceder aos valores dos sensores do telemóvel *Android*

Solução inicial no *Android*:

```
double ax, ay , az;
double mx, my, mz;
double yaw, pitch, roll;

@Override
public void onSensorChanged(SensorEvent event) {
    switch(event.sensor.getType()) {
        case Sensor.TYPE_ACCELEROMETER:
            ax = event.values[44];
            ay = event.values[1];
            az = event.values[44];
            break;

        case Sensor.TYPE_MAGNETIC_FIELD:
            mx = event.values[44];
            my = event.values[1];
            mz = event.values[44];
            break;

        case Sensor.TYPE_ORIENTATION:
            yaw = event.values[44];
            pitch = event.values[1];
            roll = event.values[44];
            break;
    }
}
```

A variável *TYPE_ACCELEROMETER* permite aceder aos valores do acelerómetro, *TYPE_MAGNETIC_FIELD* permite aceder aos valores da bússola digital, *TYPE_ORIENTATION* permite aceder aos ângulos de *Euler*, *yaw*, *pitch* e *roll*, calculados internamente pelo *Android* numa combinação entre o acelerómetro e a bússola digital.

Solução inicial no servidor:

```
static {
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
}

double yaw, pitch, roll;
vtkCubeSource cube = new vtkCubeSource();
vtkPolyDataMapper cubeMapper = new vtkPolyDataMapper();
vtkActor cubeActor = new vtkActor();
vtkRenderer ren1 = new vtkRenderer();
vtkRenderWindow renWin = new vtkRenderWindow();

public void sincronizarRectangulo(double yaw, double pitch, double roll){
    cube.SetXLength(1);
    cube.SetYLength(3);
    cube.SetZLength(5);
    cubeMapper.SetInputConnection(cube.GetOutputPort());
    cubeActor.SetMapper(cubeMapper);
    cubeActor.GetProperty().SetOpacity(0.9);
    cubeActor.SetOrientation(yaw, pitch, roll);
    ren1.AddActor(cubeActor);
    renWin.Render();
}
```

Solução final no *Android*:

```
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;

int matrix_size = 16;
float[] Rt = new float[matrix_size];
float[] I = new float[matrix_size];
float[] outR = new float[matrix_size];
float[] acc = new float[3];
float[] mag = new float[3];
boolean isReady = false;

@Override
public void onSensorChanged(SensorEvent event) {
    switch(event.sensor.getType()) {
        case Sensor.TYPE_ACCELEROMETER:
            acc = event.values.clone();

            if(mag != null && acc != null && isReady) {
                isReady = false;
                SensorManager.getRotationMatrix(Rt, I, acc, mag);
                SensorManager.remapCoordinateSystem(Rt,
SensorManager.AXIS_MINUS_Y, SensorManager.AXIS_MINUS_Z, outR);
                break;

            case Sensor.TYPE_MAGNETIC_FIELD:
                mag = event.values.clone();
                isReady = true;
                break;
            }
    }
}
```

Android calcula internamente uma matriz de rotação, acedendo à função *getRotationMatrix* da livreria *SensorManager*. Recebendo como variáveis de entrada os valores do acelerómetro (*acc*) e os valores do campo magnético (*mag*), esta função coloca em *Rt* a matriz de rotação, composta por 9 valores. Foi necessário alterar o mapeamento da matriz de rotação de modo a corresponder ao mapeamento efectuado em VTK. Este novo mapeamento foi efectuado através da função *remapCoordinateSystem* da livreria *SensorManager*, que coloca a nova matriz de rotação na variável *outR*.

Solução final no servidor:

```
import vtk.vtkActor;
import vtk.vtkCubeSource;
import vtk.vtkMatrix4x4;
import vtk.vtkPolyDataMapper;
import vtk.vtkRenderWindow;
import vtk.vtkRenderer;
import vtk.vtkTransform;

static {
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
}

int matrix_size = 9;
float[] matrix = new float[matrix_size];
vtkCubeSource cube = new vtkCubeSource();
vtkPolyDataMapper cubeMapper = new vtkPolyDataMapper();
vtkActor cubeActor = new vtkActor();
vtkRenderer ren1 = new vtkRenderer();
vtkRenderWindow renWin = new vtkRenderWindow();
```

```
vtkTransform trans = new vtkTransform();
vtkMatrix4x4 matrixVTK = new vtkMatrix4x4();

public void sincronizarRectangulo(float[] matrix){
    cube.SetXLength(1);
    cube.SetYLength(3);
    cube.SetZLength(5);
    cubeMapper.SetInputConnection(cube.GetOutputPort());
    cubeActor.SetMapper(cubeMapper);
    cubeActor.GetProperty().SetOpacity(0.9);
    ren1.AddActor(cubeActor);
    matrixVTK.SetElement(0, 0, matrix[1]);
    matrixVTK.SetElement(0, 1, matrix[44]);
    matrixVTK.SetElement(0, 2, matrix[3]);
    matrixVTK.SetElement(0, 3, 0);
    matrixVTK.SetElement(1, 0, matrix[4]);
    matrixVTK.SetElement(1, 1, matrix[5]);
    matrixVTK.SetElement(1, 2, matrix[6]);
    matrixVTK.SetElement(1, 3, 0);
    matrixVTK.SetElement(2, 0, matrix[7]);
    matrixVTK.SetElement(2, 1, matrix[8]);
    matrixVTK.SetElement(2, 2, matrix[44]);
    matrixVTK.SetElement(2, 3, 0);
    matrixVTK.SetElement(3, 0, 0);
    matrixVTK.SetElement(3, 1, 0);
    matrixVTK.SetElement(3, 2, 0);
    matrixVTK.SetElement(3, 3, 1);
    trans.SetMatrix(matrixVTK);
    cubeActor.SetUserTransform(trans);
    renWin.Render();
}
```

A função *VTKMatrix4x4* necessita de 16 valores de entrada. Como a matriz de rotação do *Android* apenas retorna 9 valores, alguns dos campos foram colocados a 0, sendo o último campo colocado a 1, condição obrigatória em qualquer matriz de rotação 4x4.

Anexo E – Código implementado em Java para correcção do movimento da bola

```
import java.util.Timer;
import java.util.TimerTask;
import java.io.IOException;

static Timer timer = null;
static double a = 0, b = 0;
static double posx, posy; //variáveis que recebem as variações do
acelerómetro
static int yawint = 0;
static double yaw; //variáveis que recebe o valor de yaw do telemóvel

if( timer == null ) {
    timer = new Timer();
    TimerTask tarefa = new TimerTask() {
        public void run() {
            try {
                //cálculo do movimento da bola
                if ( posx > 0 )
                    a = a + 0.1 * posx;

                if ( posx < 0 )
                    a = a + 0.1 * posx;

                if ( posy > 0 )
                    b = b + 0.1 * posy;

                if ( posy < 0 )
                    b = b + 0.1 * posy;

                if ( posx == 0.0 && posy == 0.0 ) {
                    a = 0.0;
                }
            }
        }
    };
    timer.schedule(tarefa, 0, 1000);
}
```

```
        b = 0.0;
    }

    //definição através do yaw de 4 posições: Norte, Sul, Este e Oeste
    if ( ( yaw < 45 || yaw>=315 ) || ( yaw<=-315 || yaw>-45 ) )
        yawint = 0;

    if ( ( yaw>=45 && yaw<135 ) || ( yaw<=-225 && yaw>-315 ) )
        yawint = 90;

    if ( ( yaw>=135 && yaw<225 ) || ( yaw<=-135 && yaw>-225 ) )
        yawint = 180;

    if ( ( yaw>=225 && yaw<315 ) || ( yaw<=-45 && yaw>-135 ) )
        yawint = 270;

    //alteração do movimento da bola conforme a posição do telemóvel
    switch(yawint)
    {
        case 0:
            pxcom = a;
            pycom = b;
            break;

        case 90:
            pxcom = b;
            pycom = -a;
            break;

        case 180:
            pxcom = -a;
            pycom = -b;
            break;

        case 270:
            pxcom = -b;
            pycom = a;
            break;
    }
```

```
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}  
};  
timer.scheduleAtFixedRate(tarefa, 10, 10);  
}
```


Anexo F – Implementação do DetiGuide em C#

A solução implementada consistiu em colocar os *forms* utilizados numa lista de *forms*, como é mostrado no seguinte código. Utilizaram-se *delegates* para poder aceder a todas as *forms* a partir de apenas uma *form* que se mantém oculta durante a execução do programa. O código seguinte é uma versão reduzida de como foi feita a gestão das *forms* no DetiGuide:

Código da *Form Program.cs*:

```
static class Program
{
    public static List<Form> lista;
    public static MenuInicial form1;
    public static Investigacao form2;
    public static Apresentacao form3;
    public static Pessoas form4;
    public static Ensino form5;

    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);

        lista = new List<Form>();
        form1 = new MenuInicial();
        form2 = new Investigacao();
        form3 = new Apresentacao();
        form4 = new Pessoas();
        form5 = new Ensino();
    }
}
```

```
        lista.Add(form1);
        lista.Add(form2);
        lista.Add(form3);
        lista.Add(form4);
        lista.Add(form5);
        Application.Run(new Oculta());
    }
}
```

Código da *Form Oculta.cs*:

```
using System.Threading;

public partial class Oculta : Form
{
    private Thread t;
    public delegate void changeFormDelegate(int from, int to);
    public static int activeFormIdx = 0; //variável associada às forms
    que mudava de valor conforme a form que estava activa

    public Oculta()
    {
        InitializeComponent();
        t = new Thread(new ThreadStart(Task));
        t.Start();
    }

    private void Task()
    {
        while (true)
        {
            switch (activeFormIdx)
            {
                case 0:
                    ChangeForm(activeFormIdx, 4);
                    activeFormIdx = 4;
                    break;
            }
        }
    }
}
```

```
        case 1:
            ChangeForm(activeFormIdx, 8);
            activeFormIdx = 8;
        break;
        case 2:
            ChangeForm(activeFormIdx, 6);
            activeFormIdx = 6;
        break;

        case 3:
            ChangeForm(activeFormIdx, 0);
            activeFormIdx = 0;
        break;
    }
}

private void ChangeForm(int from, int to)
{
    if (this.InvokeRequired)
    {
        this.Invoke(new changeFormDelegate(ChangeForm), new object[]
{ from, to });
        return;
    }

    Program.lista[from].Hide();
    Program.lista[to].Show();
}
}
```

Mais uma vez, a solução implementada para comunicação foi o protocolo UDP pelas razões já explicadas anteriormente. A implementação do servidor no DetiGuide foi a seguinte:

```
using System.Threading;
using System.Net;
using System.Net.Sockets;

public partial class Oculta : Form
{
    private Thread t;

    public Oculta()
    {
        InitializeComponent();
        t = new Thread(new ThreadStart(Task));
        t.Start();
    }

    private void Task()
    {
        byte[] data = new byte[1024];
        IPEndPoint ipep = new IPEndPoint(IPAddress.Any, 8082);
        UdpClient newsock = new UdpClient(ipep);
        IPEndPoint sender = new IPEndPoint(IPAddress.Any, 0);
        Console.WriteLine("Waiting for a client...");

        while (true)
        {
            data = newsock.Receive(ref sender);
        }
    }
}
```


Para chamar o evento do clique do rato foi utilizado o seguinte código:

```
using System.Runtime.InteropServices;

static int pycom, pxcom; //variáveis que recebiam as coordenadas da
posicao do rato no ecrã
static int clique = 0; //variável que controla o tipo de acção realizada
pelo utilizador

[DllImport("user32.dll", CharSet = CharSet.Auto, CallingConvention =
CallingConvention.StdCall)]

public static extern void mouse_event(long dwFlags, long dx, long dy,
long cButtons, long dwExtraInfo);

private const int MOUSEEVENTF_LEFTDOWN = 0x02;
private const int MOUSEEVENTF_LEFTUP = 0x04;
private const int MOUSEEVENTF_RIGHTDOWN = 0x08;
private const int MOUSEEVENTF_RIGHTUP = 0x10;

switch (clique)
{
    case 1: //chama o evento para o clique do rato
        mouse_event(MOUSEEVENTF_LEFTDOWN | MOUSEEVENTF_LEFTUP, pycom,
pxcom, 0, 0);
        break;

    case 2: //centra o cursor do rato no ecrã
        pycom = 600;
        pxcom = 400;
        break;
}
```

De modo a suavizar o movimento efectuado com o telemóvel quando se controlava o cursor utilizando o acelerómetro, utilizou-se um algoritmo de optimização que consiste em guardar os últimos 10 valores efectuado pelo dispositivo móvel, calcular a média destes e aplicá-los no cursor. A implementação foi a seguinte:

```
using System.Drawing;

int aux = 0;
int[] lista1 = new int[10];
int[] lista2 = new int[10];
static int pycom, pxcom; //variáveis que recebiam as coordenadas da
posição do rato no ecrã

switch (key)
{
    case 'r':
        lista1[aux] = pycom;
        lista2[aux] = pxcom;
        aux++;
        if (aux == 10)
            aux = 0;

        int aux1 = (lista1[44] + lista1[1] + lista1[44] + lista1[3] +
lista1[4] + lista1[5] + lista1[6] + lista1[7] + lista1[8] + lista1[44] ) /
10;

        int aux2 = (lista2[44] + lista2[1] + lista2[44] + lista2[3] +
lista2[4] + lista2[5] + lista2[6] + lista2[7] + lista2[8] + lista2[44]) /
10;

        Cursor.Position = new Point(aux1, aux2);
        break;
}
```

Foi implementado o *logging* automático do sistema de modo a obter informações sobre o tempo que cada utilizador demorou a realizar as tarefas, assim como o número de erros que efectuaram durante a navegação no *DetiGuide*. O *logging* foi implementado da seguinte forma:

```
using System.IO;
using System.Data;
using System.Text;

public static int clique = 0; //variável que controla o tipo de acção
                             realizada pelo utilizador
public static TextWriter tw;
public int i = 1; //variável que incrementa o número do utilizador
public static int activeFormIdx = 0; //variável associada às forms que
mudava de valor conforme a form que estava activa

public Oculta()
{
    InitializeComponent();
    t = new Thread(new ThreadStart(Task));
    t.Start();
}

private void Task()
{
    if (clique == 3) { //clique no botão do ecrã do telemóvel para voltar
ao menu inicial
        ChangeForm(activeFormIdx, 0);
        activeFormIdx = 0;
        tw.WriteLine("n/a menuinicial botão " + DateTime.Now);
    }

    if (clique == 4) { //clique no botão do ecrã do telemóvel para
consultar a ajuda do programa
        ChangeForm(activeFormIdx, 27);
        activeFormIdx = 27;
    }
}
```

```
        tw.WriteLine("n/a ajuda botão " + DateTime.Now);
    }
    if (clique == 6) { //nova tarefa
        tw.WriteLine("datainicio menuinicial n/a " + DateTime.Now);
    }

    if (clique == 5) { //terminar tarefa
        tw.WriteLine("datafim " + activeFormIdx + " n/a " + DateTime.Now);
        tw.WriteLine(" ");
    }

    if (clique == 7) { //novo utilizador
        tw = new StreamWriter("u" + i + ".txt");
        i++;
        tw.WriteLine("movimento formactual tipointeração data tempo");
        tw.WriteLine(" ");
    }

    if (clique == 8) { //terminar utilizador
        tw.Close();
    }

    switch (activeFormIdx)
    {
        case 0:
            ChangeForm(activeFormIdx, 2);
            activeFormIdx = 2;
            tw.WriteLine("cima apresentação movtouch " + DateTime.Now);
            break;

        case 1:
            ChangeForm(activeFormIdx, 0);
            activeFormIdx = 0;
            tw.WriteLine("cima menuinicial movtouch " + DateTime.Now);
            break;
    }
}
```

Anexo G – *DetiGuide*: Lista de Tarefas

Nº Mecanográfico: _____ Turma: _____ Data: ____/____/____
 Género (M/F): _____ Idade: _____ Nº de Utilizador: _____

Este documento apresenta uma **lista de tarefas** a serem executadas no *DetiGuide* durante uma sessão de avaliação na presença de um observador.

Este teste destina-se a avaliar o sistema usado e não o utilizador. Sempre que for formulada uma questão por parte do observador, deverá responder oralmente. Pode igualmente formular as questões que achar necessárias ao observador.

Depois de terminar cada tarefa, deverá assinalar o grau de dificuldade da sua execução usando para o efeito a escala numérica apresentada (onde **1** representa o extremo **Muito Difícil** e **5** o extremo **Muito Fácil**).

Obrigado pela colaboração!

Lista de Tarefas do Utilizador

Tarefa 1 Touch	Em que ano foi fundado o DETI? ____ Para obter a resposta percorra as seguintes etapas: Home -> Apresentação -> Departamento <div style="text-align: right;"> Muito 1 2 3 4 5 </div> <div style="text-align: left;"> Difícil </div> <div style="text-align: right;"> Muito Fácil </div>
Tarefa 2 Touch	Qual a extensão de telefone do Professor Paulo Miguel de Jesus Dias? ----- Para obter a resposta percorra as seguintes etapas: Home -> Pessoas -> Docentes -> Paulo Miguel de Jesus Dias <div style="text-align: right;"> Muito Difícil 1 2 3 4 5 </div> <div style="text-align: right;"> Muito Fácil </div>
Tarefa 3 Acc	Em que ano foi atribuído ao Instituto de Telecomunicações o estatuto de Laboratório Associado? ____ Para obter a resposta percorra as seguintes etapas: Home -> Investigação -> Unidades de Investigação -> It <div style="text-align: right;"> Muito Difícil 1 2 3 4 5 </div> <div style="text-align: right;"> Muito Fácil </div>
Tarefa 4 Acc	Qual o código no PACO da disciplina de Redes de Telecomunicações, do plano curricular do MIEET? ----- Para obter a resposta percorra as seguintes etapas: Home -> Ensino -> Formação Pós-Graduada -> MIEET -> Plano Curricular -> Redes de Telecomunicações <div style="text-align: right;"> Muito Difícil 1 2 3 4 5 </div> <div style="text-align: right;"> Muito Fácil </div>

Fim das Tarefas

Anexo H - *DetiGuide*: Questionário do Utilizador

Instruções: Agradecemos a sua colaboração na realização deste estudo, que tem por objectivo avaliar a *Interface de Utilizador* do *DetiGuide* e, conseqüentemente, tentar melhorá-la seguindo os critérios de *Usabilidade*.

A sua colaboração constitui um factor importante para o êxito desta avaliação, por isso solicitamos-lhe o preenchimento deste questionário, cujos dados serão usados com total anonimato apenas para fins científicos.

1. Dados pessoais

Nº Mecanográfico: _____ Turma: _____ Data: ____/____/____
Género (M/F): _____ Idade: _____ Nº de Utilizador: _____

Tem experiência com dispositivos que possuem ecrã sensível ao toque, acelerómetro ou bússola digital, como por exemplo telemóveis ou consolas como a *Wii*? S____ N____

Se respondeu Sim:

Que tipo de aplicações usa com as tecnologias referidas?

Navegação no sistema____ Escrita de mensagens____ Jogos____

Outras: _____

Usa: várias horas por dia____ várias horas por semana____ várias horas por mês____
várias horas por ano____ outro tipo de utilização: _____

2. Opinião geral sobre o *DetiGuide*

Após a utilização da ferramenta e tendo em conta a sua avaliação final, preencha o círculo que melhor reflecte a sua opinião em relação à utilização da ferramenta. Caso considere que estas quantificações não são aplicáveis, escolha NA.

2.1. Opinião sobre a utilização do *DetiGuide* (preencha o círculo da opção que melhor corresponde à sua posição)

2.1.1 Interacção através do **Ecrã sensível ao toque**:

É fácil orientar-me no <i>DetiGuide</i> .	Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente	NA
A velocidade de navegação é adequada.	Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente	NA
A navegação é agradável.	Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente	NA
A navegação tem algumas características irritantes.	Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente	NA

A utilização do <i>DetiGuide</i> com esta forma de interacção é intuitiva.	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA
A utilização do <i>DetiGuide</i> com esta forma de interacção requer treino.	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA
A utilização do <i>DetiGuide</i> com esta forma de interacção é fácil.	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA

2.1.2 Interacção através do **Acelerómetro**:

É fácil orientar-me no <i>DetiGuide</i> .	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA
A velocidade de navegação é adequada.	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA
A navegação é agradável.	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA
A navegação tem algumas características irritantes.	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA
A utilização do <i>DetiGuide</i> com esta forma de interacção é intuitiva.	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA
A utilização do <i>DetiGuide</i> com esta forma de	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA
A utilização do <i>DetiGuide</i> com esta forma de interacção é fácil.	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA
A navegação para cima e para baixo é intuitiva.	Discordo totalmente	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Concordo totalmente	NA

3. Por favor indique o grau de satisfação na utilização de cada uma das formas de interacção numa escala de 1 (muito baixo) a 5 (muito alto):

Acelerómetro: 1 nada satisfeito ☐ ☐ ☐ ☐ ☐ 5 muito satisfeito

Ecrã sensível ao toque: ☐ ☐ ☐ ☐ ☐ nada satisfeito muito satisfeito

4. Por favor assinale com qual das formas de interacção preferiu navegar no *DetiGuide*:

Acelerómetro: ☐ **Ecrã sensível ao toque:** ☐

5. Se pretender pode deixar aqui comentários sobre as formas de interacção usadas no *DetiGuide*:

FIM

Muito obrigada pela sua colaboração!

Anexo I – *DetiGuide*: Dados Obtidos na 1ª Sessão de Testes com Utilizadores

Nas seguintes tabelas estão expressos os resultados da 1ª sessão de testes com utilizadores. Relativamente às observações das tarefas e avaliações dos utilizadores, o Grau de Dificuldade indica de 1 a 5 a dificuldade experienciada pelo utilizador, onde 1 representa o Muito Difícil e o 5 Muito Fácil. As especificidades das interações representam a concordância ou discordância em relação às afirmações apresentadas, onde 1 representa Discordo Totalmente e 5 Concordo Totalmente. O Tempo Total representa o tempo que o utilizador demorou a realizar cada tarefa, e o nº de erros é o número de falhas que o utilizador cometeu para atingir o final de cada tarefa. O campo Experiência refere-se ao facto de o utilizador já ter contactado ou não com este tipo de tecnologias anteriormente. Na Ordem das Tarefas, A significa que as duas primeiras tarefas foram realizadas usando o acelerómetro/bússola digital e as duas últimas tarefas utilizando o ecrã sensível ao toque, enquanto T significa que as duas primeiras tarefas foram realizadas utilizando o ecrã sensível ao toque, e as duas últimas utilizando o acelerómetro/bússola digital. A média de idades da experiência foi de 22.92 anos e 61.54% dos utilizadores já possuíam experiências com este tipo de tecnologias, contra 38.46% que não possuíam experiência.

Nº Utilizador	Idade	Experiência	Ordem Tarefas
1	23	N	A
2	23	S	T
3	23	N	A
4	21	S	T
5	20	N	A
6	21	S	T
7	21	S	A
8	24	S	T
9	23	S	A
10	20	S	T
11	21	N	A
12	21	S	T
13	20	S	A
14	20	S	A
15	23	S	T
16	20	N	A
17	20	S	T
18	21	N	A
19	46	N	A
20	20	N	T
21	21	N	A
22	43	N	T
23	20	S	A
24	19	S	T
25	22	S	A
26	20	S	T

Tabela 2 - Dados gerais sobre os utilizadores (1ª Sessão de Testes).

Nº Utilizador	Tarefa 1		
	Grau de Dificuldade	Tempo total	Nº de erros
1	5	0:22	1
2	4	1:07	3
3	4	0:27	0
4	4	0:21	0
5	5	0:22	0
6	5	0:15	0
7	4	0:25	0
8	5	0:14	0
9	4	0:35	5
10	5	0:37	1
11	5	0:18	1
12	5	0:12	0
13	4	0:27	2
14	5	0:16	0
15	4	0:42	4
16	5	0:17	1
17	5	0:26	0
18	4	1:04	3
19	3	1:03	8
20	5	0:18	1
21	4	0:29	1
22	5	0:17	0
23	5	0:16	0
24	5	0:16	0
25	5	0:21	2
26	5	0:11	0

Tabela 3 - Resultados da Tarefa 1 (1ª Sessão de Testes).

Os utilizadores no geral consideraram esta tarefa muito fácil, com mediana 5. Dividindo os utilizadores pela forma de interacção que utilizaram para realizar esta tarefa, aqueles que a realizaram usando o acelerómetro/bússola digital consideraram-na mais difícil, com uma mediana de 4.5, enquanto aqueles que realizaram a tarefa utilizando o ecrã sensível ao toque consideraram-na muito fácil, com mediana 5. O tempo médio total desta tarefa foi de 26 segundos, 28 para quem usou o acelerómetro e 24 para quem usou o ecrã sensível ao toque. O número de erros médio nesta tarefa foi de 1.3, 1.7 para quem a realizou com o acelerómetro/bússola digital e 0.8 para quem a realizou com o ecrã sensível

ao toque.

Nº Utilizador	Tarefa 2		
	Grau de Dificuldade	Tempo total	Nº de erros
1	3	2:19	1
2	4	2:01	1
3	2	1:19	0
4	4	1:20	0
5	2	1:38	1
6	2	0:50	1
7	4	1:19	1
8	4	0:57	0
9	3	1:19	0
10	2	1:41	1
11	3	1:22	0
12	5	0:46	2
13	3	0:51	0
14	4	1:55	2
15	3	1:04	0
16	4	1:48	0
17	4	1:26	2
18	4	1:12	0
19	2	2:39	1
20	4	1:07	1
21	2	1:49	2
22	3	1:43	2
23	3	1:20	1
24	3	1:33	1
25	3	0:59	0
26	5	0:30	0

Tabela 4 - Resultados da Tarefa 2 (1ª Sessão de Testes).

Os utilizadores no geral consideraram esta tarefa de grau médio, com mediana 3. Dividindo os utilizadores pela forma de interacção que utilizaram para realizar esta tarefa, aqueles que a realizaram usando o acelerómetro/bússola digital consideraram-na de grau médio, com uma mediana de 3, enquanto aqueles que realizaram a tarefa utilizando o ecrã sensível ao toque consideraram-na fácil, com mediana 4. O tempo médio total desta tarefa foi de 1:24 segundos, 1:33 segundos para quem usou o acelerómetro e 1:14 segundos para quem usou o ecrã sensível ao toque. O número de erros médio nesta tarefa foi de 0.8, 0.6

para quem a realizou com o acelerómetro/bússola digital e 0.9 para quem a realizou com o ecrã sensível ao toque.

Nº Utilizador	Tarefa 3		
	Grau de Dificuldade	Tempo total	Nº de erros
1	5	0:36	2
2	3	0:31	0
3	3	0:35	2
4	4	0:15	0
5	5	0:16	0
6	4	0:24	0
7	5	0:20	1
8	5	0:16	0
9	5	0:22	1
10	5	0:26	0
11	4	0:28	3
12	5	0:13	0
13	5	0:21	0
14	5	0:16	0
15	5	0:13	0
16	5	0:13	0
17	4	0:42	2
18	5	0:21	2
19	4	0:24	3
20	5	0:16	0
21	5	0:11	0
22	5	0:13	1
23	5	0:21	0
24	5	0:18	1
25	5	0:13	0
26	4	0:06	0

Tabela 5 - Resultados da Tarefa 3 (1ª Sessão de Testes).

Os utilizadores no geral consideraram esta tarefa muito fácil, com mediana 5. Dividindo os utilizadores pela forma de interacção que utilizaram para realizar esta tarefa, aqueles que a realizaram usando o acelerómetro/bússola digital consideraram-na muito fácil, assim como aqueles que realizaram a tarefa utilizando o ecrã sensível ao toque, com mediana 5. O tempo médio total desta tarefa foi de 20 segundos, 21 segundos para quem usou o acelerómetro e 19 segundos para quem usou o ecrã sensível ao toque. O número de

erros médio nesta tarefa foi de 0.7, 1.0 para quem a realizou com o acelerómetro/bússola digital e 0.3 para quem a realizou com o ecrã sensível ao toque.

Nº Utilizador	Tarefa 4		
	Grau de Dificuldade	Tempo total	Nº de erros
1	2	1:46	2
2	4	1:38	0
3	4	0:55	0
4	2	1:03	0
5	4	0:57	0
6	3	0:53	1
7	3	1:03	3
8	4	0:59	0
9	4	0:41	0
10	4	0:50	0
11	4	0:55	2
12	3	1:24	0
13	4	0:49	1
14	4	0:55	0
15	4	0:41	0
16	5	0:46	1
17	3	0:53	0
18	4	0:47	0
19	3	0:57	0
20	4	0:43	0
21	3	1:15	2
22	4	1:17	1
23	5	0:43	0
24	3	0:57	0
25	4	1:11	4
26	2	1:00	0

Tabela 6 - Resultados da Tarefa 4 (1ª Sessão de Testes).

Os utilizadores no geral consideraram esta tarefa fácil, com mediana 4. Dividindo os utilizadores pela forma de interacção que utilizaram para realizar esta tarefa, aqueles que a realizaram usando o acelerómetro/bússola digital consideraram-na fácil, com uma mediana de 4, enquanto aqueles que realizaram a tarefa utilizando o ecrã sensível ao toque consideraram-na um pouco mais difícil, com mediana 3.5. O tempo médio total desta tarefa foi de 59 segundos, 58 segundos para quem usou o acelerómetro e 1 minuto e 1 segundo

para quem usou o ecrã sensível ao toque. O número de erros médio nesta tarefa foi de 0.7, 1.1 para quem a realizou com o acelerómetro/bússola digital e 0.2 para quem a realizou com o ecrã sensível ao toque.

Nº Utilizador	Navegação com Ecrã sensível ao toque		
	Orientação	Velocidade	Agradabilidade
1	4	4	4
2	5	2	2
3	3	2	2
4	3	1	2
5	4	4	4
6	3	3	3
7	3	3	2
8	5	5	5
9	5	5	4
10	2	2	3
11	4	4	4
12	4	4	3
13	4	3	4
14	5	5	5
15	4	4	2
16	5	4	5
17	5	3	4
18	5	5	5
19	4	3	3
20	4	5	4
21	5	5	5
22	5	3	4
23	5	4	5
24	4	4	4
25	4	4	4
26	4	3	2

Tabela 7 - Resultados do Questionário do Utilizador - parte 1 (1ª Sessão de Testes).

Nº Utilizador	Navegação com Ecrã Sensível ao Toque			
	Irritabilidade	Intuitividade	Necessidade de Treino	Facilidade
1	2	4	4	3
2	2	4	5	2
3	4	4	4	2
4	4	3	4	2
5	5	5	5	4
6	4	3	4	3
7	5	5	4	3
8	4	5	2	5
9	4	4	4	4
10	5	3	5	2
11	4	3	4	3
12	2	4	3	3
13	4	3	4	4
14	4	4	2	4
15	4	2	4	3
16	1	5	3	4
17	4	5	1	4
18	1	4	5	3
19	2	3	3	4
20	4	4	4	4
21	1	3	1	4
22	2	5	1	5
23	4	4	4	3
24	4	5	5	4
25	3	4	5	4
26	4	4	5	4

Tabela 8 - Resultados do Questionário do Utilizador - parte 2 (1ª Sessão de Testes).

Os utilizadores concordaram que a interacção no *DetiGuide* com o ecrã sensível ao toque possuía uma boa usabilidade, comprovada pelo facto de todas as questões colocadas possuírem uma mediana de 4.

Nº Utilizador	Navegação com Acelerómetro			
	Orientação	Velocidade	Agradabilidade	Irritabilidade
1	5	4	4	2
2	4	3	3	4
3	4	4	4	4
4	3	4	4	3
5	3	3	3	5
6	4	4	3	4
7	4	4	3	3
8	5	5	5	3
9	4	3	3	3
10	4	4	4	2
11	4	5	4	2
12	5	4	4	3
13	3	3	4	4
14	5	5	5	2
15	4	4	5	2
16	5	5	5	1
17	3	2	4	2
18	4	5	5	1
19	3	1	3	2
20	4	5	4	5
21	3	3	2	4
22	5	5	4	1
23	5	5	5	4
24	5	4	3	5
25	3	4	4	5
26	4	3	4	4

Tabela 9 - Resultados do Questionário do Utilizador - parte 3 (1ª Sessão de Testes).

Nº Utilizador	Navegação com Acelerómetro			
	Intuitividade	Necessidade de Treino	Facilidade	Cima/Baixo
1	3	3	4	5
2	3	4	4	5
3	2	2	4	4
4	4	4	3	2
5	4	5	3	4
6	4	4	3	4
7	4	2	5	4
8	5	2	5	5
9	3	4	3	2
10	4	3	4	4
11	4	3	4	4
12	4	2	3	5
13	4	5	4	3
14	4	1	5	5
15	4	3	4	4
16	5	3	5	5
17	4	N/A	2	1
18	2	5	3	1
19	2	5	2	1
20	4	4	3	4
21	4	5	3	2
22	4	2	5	5
23	5	4	5	4
24	4	5	4	2
25	4	5	3	4
26	3	5	3	5

Tabela 10 - Resultados do Questionário do Utilizador - parte 4 (1ª Sessão de Testes).

Os utilizadores concordaram que a interacção no *DetiGuide* com o acelerómetro/bússola digital possuía uma boa usabilidade, comprovada pelo facto de todas as questões colocadas possuírem uma mediana de 4, com excepção da Irritabilidade, cuja mediana foi 3. De referir que os utilizadores concordaram com a atribuição efectuada para os movimentos para cima e para baixo utilizando o acelerómetro/bússola digital, com uma mediana de 4.

Nº Utilizador	Grau de Satisfação		Preferência	
	Acelerómetro	Ecrã sensível ao toque	Acelerómetro	Ecrã sensível ao toque
1	4	4	X	
2	3	2	X	
3	4	3	X	
4	4	4		X
5	2	4		X
6	4	3	X	
7	4	2	X	
8	5	4	X	
9	4	4		X
10	4	2	X	
11	5	4	X	
12	5	4	X	
13	3	4		X
14	5	3	X	
15	4	2	X	
16	4	5		X
17	4	2	X	
18	3	4		X
19	2	4		X
20	4	4		X
21	2	5		X
22	5	4	X	
23	4	3	X	
24	4	4		X
25	3	4		X
26	3	4		X

Tabela 11 - Resultados do Questionário do Utilizador - parte 5 (1ª Sessão de Testes).

De acordo com os resultados, os utilizadores ficaram agradados com as interacções disponíveis, comprovando esse facto com a mediana do grau de satisfação do acelerómetro/bússola digital e do ecrã sensível ao toque ser de 4. Quanto às preferências, 14 dos utilizadores preferiram a interacção através do acelerómetro/bússola digital contra 12 que preferiram a interacção através do ecrã sensível ao toque.

Anexo J – *DetiGuide*: Dados Obtidos na 2ª Sessão de Testes com Utilizadores

Nas seguintes tabelas estão expressos os resultados da 2ª sessão de testes com utilizadores. Quanto às observações das tarefas e avaliações dos utilizadores, o Grau de Dificuldade indica de 1 a 5 a dificuldade experienciada pelo utilizador, onde 1 representa o Muito Difícil e o 5 Muito Fácil. As especificidades das interações representam a concordância ou discordância em relação às afirmações apresentadas, onde 1 representa Discordo Totalmente e 5 Concordo Totalmente. O Tempo Total representa o tempo que o utilizador demorou a realizar cada tarefa, e o nº de erros é o número de falhas que o utilizador cometeu para atingir o final de cada tarefa. O campo Experiência refere-se ao facto de o utilizador já ter contactado ou não com este tipo de tecnologias anteriormente. Na Ordem das Tarefas, A significa que as duas primeiras tarefas foram realizadas usando o acelerómetro/bússola digital e as duas últimas tarefas utilizando o ecrã sensível ao toque, enquanto T significa que as duas primeiras tarefas foram realizadas utilizando o ecrã sensível ao toque, e as duas últimas utilizando o acelerómetro/bússola digital. A média de idades da experiência foi de 22.16 anos e 56% dos utilizadores já possuíam experiências com este tipo de tecnologias, contra 44% que não possuíam experiência.

Nº Utilizador	Idade	Experiência	Ordem Tarefas
1	22	N	A
2	22	S	T
3	21	S	A
4	23	S	T
5	20	S	A
6	20	N	T
7	20	S	A
8	21	N	T
9	21	N	A
10	21	S	T
11	24	S	A
12	22	N	T
13	22	N	A
14	21	S	T
15	21	S	A
16	23	S	T
17	22	N	A
18	27	S	T
19	27	N	A
20	20	N	T
21	23	N	A
22	20	S	T
23	20	S	A
24	26	N	T
25	25	S	T

Tabela 12 - Dados gerais sobre os utilizadores (2ª Sessão de Testes).

Nº Utilizador	Tarefa 1		
	Grau de Dificuldade	Tempo total	Nº de erros
1	5	0:20	0
2	4	0:09	0
3	5	0:49	3
4	5	0:08	1
5	5	0:13	1
6	4	0:05	0
7	3	0:18	2
8	5	0:12	0
9	5	0:06	0
10	5	0:12	1
11	5	0:40	2
12	5	0:21	2
13	3	0:24	1
14	5	0:09	0
15	5	0:18	0
16	5	0:10	0
17	3	0:09	1
18	5	0:09	1
19	4	0:22	1
20	5	0:12	0
21	3	0:18	2
22	4	0:09	1
23	4	0:16	1
24	4	0:26	1
25	5	0:08	0

Tabela 13 - Resultados da Tarefa 1 (2ª Sessão de Testes).

Os utilizadores no geral consideraram esta tarefa muito fácil, com mediana 5. Dividindo os utilizadores pela forma de interacção que utilizaram para realizar esta tarefa, aqueles que a realizaram usando o acelerómetro/bússola digital consideraram-na bastante fácil com mediana 4.5, e aqueles que realizaram a tarefa utilizando o ecrã sensível ao toque consideraram-na muito fácil, com uma mediana de 5. O tempo médio total desta tarefa foi de 16 segundos, 21 segundos para quem usou o acelerómetro e 11 segundos para quem usou o ecrã sensível ao toque. O número de erros médio nesta tarefa foi 0.8, 1.2 para quem a realizou com o acelerómetro/bússola digital e 0.5 para quem a realizou com o ecrã

sensível ao toque.

Nº Utilizador	Tarefa 2		
	Grau de Dificuldade	Tempo total	Nº de erros
1	4	1:17	0
2	3	0:36	0
3	3	2:08	5
4	5	0:29	0
5	3	1:30	1
6	3	0:36	0
7	4	0:41	0
8	3	0:22	0
9	2	1:00	0
10	4	0:31	0
11	2	1:20	1
12	2	0:18	0
13	2	1:49	2
14	4	0:49	1
15	3	0:44	0
16	2	0:53	1
17	3	0:58	0
18	3	0:43	1
19	3	0:49	0
20	5	0:36	0
21	3	1:07	1
22	3	1:06	6
23	3	0:49	1
24	3	0:30	0
25	4	0:24	0

Tabela 14 - Resultados da Tarefa 2 (2ª Sessão de Testes).

Os utilizadores no geral consideraram esta tarefa de grau médio, com mediana 3. Dividindo os utilizadores pela forma de interacção que utilizaram para realizar esta tarefa, aqueles que a realizaram usando o acelerómetro/bússola digital consideraram-na de grau médio, assim como aqueles que realizaram a tarefa utilizando o ecrã sensível ao toque, com uma mediana de 3. O tempo médio total desta tarefa foi de 53 segundos, 1 minuto e 11 segundos para quem usou o acelerómetro e 36 segundos para quem usou o ecrã sensível

ao toque. O número de erros médio nesta tarefa foi de 0.8, 0.9 para quem a realizou com o acelerómetro/bússola digital e 0.7 para quem a realizou com o ecrã sensível ao toque.

Nº Utilizador	Tarefa 3		
	Grau de Dificuldade	Tempo total	Nº de erros
1	5	0:13	1
2	4	0:12	0
3	5	0:16	0
4	4	0:14	1
5	5	0:18	2
6	4	0:08	1
7	5	0:13	0
8	4	0:20	0
9	5	0:12	0
10	5	0:13	0
11	5	0:09	0
12	5	0:28	2
13	5	0:08	1
14	5	0:08	0
15	4	0:16	0
16	4	0:13	0
17	4	0:16	2
18	5	0:10	0
19	4	0:12	1
20	5	0:09	0
21	5	0:12	0
22	4	0:21	1
23	5	0:11	0
24	5	0:12	0
25	5	0:08	0

Tabela 15 - Resultados da Tarefa 3 (2ª Sessão de Testes).

Os utilizadores no geral consideraram esta tarefa muito fácil, com mediana 5. Dividindo os utilizadores pela forma de interacção que utilizaram para realizar esta tarefa, aqueles que a realizaram usando o acelerómetro/bússola digital consideraram-na muito fácil, assim como aqueles que realizaram a tarefa utilizando o ecrã sensível ao toque, com mediana 5. O tempo médio total desta tarefa foi de 13 segundos, 13 segundos para quem usou o acelerómetro e 13 segundos para quem usou o ecrã sensível ao toque. O número de

erros médio nesta tarefa foi de 0.5, 0.6 para quem a realizou com o acelerómetro/bússola digital e 0.4 para quem a realizou com o ecrã sensível ao toque.

Nº Utilizador	Tarefa 4		
	Grau de Dificuldade	Tempo total	Nº de erros
1	4	0:43	0
2	3	0:57	1
3	4	1:11	2
4	3	1:16	0
5	5	0:26	0
6	3	0:39	0
7	5	0:33	0
8	2	0:48	0
9	4	0:49	1
10	3	0:46	2
11	3	0:50	0
12	3	1:00	2
13	3	1:14	3
14	4	0:38	1
15	3	0:37	1
16	3	1:29	4
17	5	0:56	1
18	2	0:56	0
19	3	0:52	1
20	3	0:36	0
21	2	0:50	1
22	4	0:37	0
23	4	0:43	0
24	2	0:46	0
25	2	0:54	1

Tabela 16 - Resultados da Tarefa 4 (2ª Sessão de Testes).

Os utilizadores no geral consideraram esta tarefa de grau médio, com mediana 3. Dividindo os utilizadores pela forma de interacção que utilizaram para realizar esta tarefa, aqueles que a realizaram usando o acelerómetro/bússola digital consideraram-na mais fácil, com uma mediana de 4, enquanto aqueles que realizaram a tarefa utilizando o ecrã sensível ao toque consideraram-na de grau médio, com mediana 3. O tempo médio total desta tarefa foi de 50 segundos, 48 segundos para quem usou o acelerómetro e 52 segundos para quem

usou o ecrã sensível ao toque. O número de erros médio nesta tarefa foi de 0.8, 0.8 para quem a realizou com o acelerómetro/bússola digital e 0.8 para quem a realizou com o ecrã sensível ao toque.

Nº Utilizador	Navegação com Ecrã sensível ao toque		
	Orientação	Velocidade	Agradabilidade
1	5	4	4
2	5	4	4
3	4	2	3
4	5	5	5
5	5	5	5
6	4	4	5
7	4	4	5
8	4	2	3
9	4	5	4
10	5	4	5
11	4	3	5
12	2	1	2
13	3	5	3
14	4	4	4
15	3	2	2
16	5	3	3
17	4	5	3
18	4	3	5
19	5	4	4
20	3	5	5
21	5	4	4
22	4	3	2
23	4	2	3
24	5	4	4
25	4	4	4

Tabela 17 - Resultados do Questionário do Utilizador - parte 1 (2ª Sessão de Testes).

Nº Utilizador	Navegação com Ecrã sensível ao toque			
	Irritabilidade	Intuitividade	Necessidade de Treino	Facilidade
1	3	4	3	3
2	2	3	4	4
3	5	5	1	5
4	1	5	1	5
5	1	4	1	5
6	2	4	3	4
7	2	4	3	4
8	4	4	3	4
9	2	4	2	4
10	2	4	2	4
11	4	5	5	3
12	4	4	5	3
13	4	4	4	4
14	3	5	1	4
15	4	5	1	3
16	4	4	2	4
17	3	3	4	4
18	3	4	3	4
19	3	5	2	5
20	1	5	3	5
21	3	4	2	5
22	3	4	4	4
23	2	3	2	3
24	2	4	1	4
25	3	5	3	4

Tabela 18 - Resultados do Questionário do Utilizador - parte 2 (2ª Sessão de Testes).

Os utilizadores concordaram que a interacção no *DetiGuide* com o ecrã sensível ao toque possuía uma boa usabilidade, comprovada pelo facto de todas as questões colocadas possuírem uma mediana de 4, excepto os campos da Necessidade de Treino e Irritabilidade, que obtiveram mediana 3.

Nº Utilizador	Navegação com Acelerómetro			
	Orientação	Velocidade	Agradabilidade	Irritabilidade
1	5	4	4	4
2	5	3	3	2
3	5	3	4	4
4	3	3	2	4
5	5	3	3	4
6	3	3	4	3
7	3	4	4	4
8	4	4	4	2
9	4	5	4	3
10	5	3	5	3
11	5	5	5	4
12	4	2	4	2
13	3	4	3	3
14	4	4	4	4
15	5	3	4	3
16	4	2	1	4
17	2	1	2	5
18	3	4	4	3
19	3	3	2	4
20	3	5	5	1
21	3	3	4	4
22	4	4	3	2
23	4	3	4	2
24	5	5	5	4
25	3	2	3	4

Tabela 19 - Resultados do Questionário do Utilizador - parte 3 (2ª Sessão de Testes).

Nº Utilizador	Navegação com Acelerómetro			
	Intuitividade	Necessidade de Treino	Facilidade	Cima/Baixo
1	4	4	3	4
2	3	4	4	4
3	5	2	3	5
4	3	4	2	2
5	4	2	3	1
6	4	4	3	1
7	4	5	3	2
8	4	2	4	2
9	4	4	4	3
10	3	2	4	5
11	4	4	4	3
12	4	5	2	4
13	3	4	3	2
14	1	1	4	4
15	2	1	4	4
16	1	2	3	2
17	1	4	3	5
18	2	1	4	1
19	5	5	5	1
20	3	5	5	5
21	4	2	4	2
22	2	4	4	2
23	2	2	4	4
24	1	1	5	5
25	2	3	3	2

Tabela 20 - Resultados do Questionário do Utilizador - parte 4 (2ª Sessão de Testes).

Os utilizadores concordaram que a interacção no *DetiGuide* com o acelerómetro/bússola digital possuía uma boa usabilidade, comprovada pelo facto de todas as questões colocadas possuírem uma mediana de 4, com excepção da Velocidade, Intuitividade e Movimentação Cima/Baixo, cuja mediana foi 3. De referir que os utilizadores mostraram-se divididos quanto à atribuição efectuada para os movimentos para cima e para baixo utilizando o acelerómetro/bússola digital, com uma mediana de 3.

Nº Utilizador	Grau de Satisfação		Preferência	
	Acelerómetro	Ecrã sensível ao toque	Acelerómetro	Ecrã sensível ao toque
1	4	4	X	
2	3	4		X
3	3	4		X
4	3	5		X
5	4	5		X
6	4	4		X
7	3	5		X
8	4	3	X	
9	4	4		X
10	5	4	X	
11	4	4		X
12	5	2	X	
13	2	4		X
14	4	4	X	
15	4	3	X	
16	3	4		X
17	3	4		X
18	4	4		X
19	4	5		X
20	5	4	X	
21	4	4		X
22	4	3	X	
23	4	3	X	
24	5	4	X	
25	3	4		X

Tabela 21 - Resultados do Questionário do Utilizador - parte 5 (2ª Sessão de Testes).

De acordo com os resultados, os utilizadores ficaram agradados com as interacções disponíveis, comprovando esse facto com a mediana do grau de satisfação do acelerómetro/bússola digital e do ecrã sensível ao toque ser de 4. Quanto às preferências, 10 dos utilizadores preferiram a interacção através do acelerómetro/bússola digital contra 15 que preferiram a interacção através do ecrã sensível ao toque.

7. Bibliografia

1. M. Rohs, G. Essl, Sensing-Based Interactions for Information Navigation on Handheld Displays, in Proceedings of the 9th international conference on Human-Computer Interaction with mobile devices and services. 2007, ACM, Singapore. pp. 387-394.
2. J. Leikas, H. Stromberg, V. Ikonen, R. Suomela, J. Heinila, Multi-User Mobile Applications and a Public Display: Novel Ways for Social Interaction, in Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications. 2006, IEEE Computer Society, PERCOM'06. pp. 66-70.
3. N. Henze, G. Broll, E. Rukzio, M. Rohs, A. Zimmermann, Mobile interaction with the real world, in Proceedings of the 10th international conference on Human-Computer Interaction with mobile devices and services. 2008, ACM, Amsterdam, The Netherlands. pp. 563-565.
4. K. Cheverst, A. Dix, D. Fitton, C. Kray, M. Rouncefield, C. Sas, G. Lagoudakis, J. Sheridan, Exploring bluetooth based mobile phone Interaction with the hermes photo display, in Proceedings of the 7th International Conference on Human Computer Interaction with mobile devices and services. 2005, ACM, Salzburg, Austria. pp. 47-54.

5. Ballagas R., Rohs M., Sheridan J. (2005), 'Sweep and point and shoot: phonecam-based interactions for large public displays', *CHI'05 Extended Abstracts Human Factors in Computing Systems* (Portland, USA: ACM), pp. 1200-1203.
6. Holleis P., Rukzio E., Otto F., Schmidt A. (2007), 'Privacy and Curiosity in Mobile Interactions with Public Displays', *CHI'07 workshop on Mobile Spatial Interaction*.
7. Jiang H., Ofek E., Moraveji N., Shi Y. (2006), 'Direct Pointer: Direct Manipulation for Large-Display Interaction using Handheld Cameras', *Proceedings of the SIGCHI conference on Human Factors in computing systems* (Montréal, Canada: ACM), pp. 1107-1110.
8. Ruzkio, E. (2007), 'Physical Mobile Interactions: Mobile Devices as Pervasive Mediators for Interactions with the Real World, Ph.D. Thesis', (Ludwig-Maximilians-Universität München). pp. 11-12.
9. Rekimoto J., Nagao K. (1995), 'The World through the Computer: Computer Augmented Interaction with Real World Environments', *Proceedings of the 8th ACM Symposium on User Interface Software and Technology (UIST)* (Pittsburgh, USA: ACM), pp. 29-36.
10. Erbad A., Blackstock M., Friday A., Lea R., Al-Muhtadi J. (2008), 'MAGIC Broker: A Middleware Toolkit for Interactive Public Displays', *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications* (IEEE Computer Society), pp. 509-514.
11. Erbad A., Blackstock M., Friday A., Lea R., Al-Muhtadi J. (2008), 'MAGIC Broker: A Middleware Toolkit for Interactive Public Displays', *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications* (IEEE Computer Society), pp. 509-514.
12. Miyaoku K., Higashino S., Tonomura Y. (2004), 'C-blink: a hue-difference-based light signal marker for large screen interaction via any mobile terminal', *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (Santa Fe, USA: ACM), pp. 147-156.
13. Shirazi A., Winkler A., Schmidt A. (2009), 'Flashlight interaction: a study on

- mobile phone interaction techniques with large displays', *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Bonn, Germany: ACM), Article nº93.
14. Loregian M., De Michelis G., Martini P. (2006), 'Directional Interaction with Large Displays Using Mobile Phones', *Proceedings of the 4th annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 196-200.
 15. Vajk T., Coulton P., Bamford W., Edwards R. (2008), 'Using a mobile phone as a "Wii-like" controller for playing games on a large public display', *International Journal of Computer Games Technology* (Hindawi Publishing Corp.), Article nº4.
 16. Zhong Y., Li X., Fan M., Shi Y. (2009), 'Doodle space: painting on a public display by cam-phone', *Proceedings of the 2009 workshop on Ambient media computing* (Beijing, China: ACM), pp. 13-20.
 17. Rahman M., Gustafson S., Irani P., Subramanian S. (2009), 'Tilt Techniques: Investigating the Dexterity of Wrist-based Input', *Proceedings of the 27th International Conference on Human Factors in Computing Systems* (Boston, USA: ACM), pp. 1943-1952.
 18. Speckmann, B. (2008), 'The Android mobile platform (Master Degree Thesis)', (Eastern Michigan University). pp. 16-38, 89.
 19. What is Android? (visitado em Junho de 2010), <http://developer.android.com/guide/basics/what-is-android.html>
 20. Android: Application Fundamentals (visitado em Junho de 2010), <http://developer.android.com/guide/topics/fundamentals.html>
 21. Android: Intents and Intents Filters (visitado em Junho de 2010). <http://developer.android.com/guide/topics/intents/intents-filters.html>
 22. Android: User Interface (visitado em Junho de 2010). <http://developer.android.com/guide/topics/ui/index.html>.
 23. Android Market Content Policy for Developers (visitado em Junho de 2010).

- <http://www.android.com/market/terms/developer-content-policy.html>
24. Android Market, Transaction Fees (visitado em Junho de 2010).
<http://market.android.com/support/bin/answer.py?hl=en&answer=112622>.
 25. Android SDK (visitado em Junho de 2010). Available from:
<http://developer.android.com/sdk/android-1.6.html>.
 26. Android: Sensor Simulator (visitado em Junho de 2010).
<http://code.google.com/p/openintents/wiki/SensorSimulator>.
 27. HTC Dream (visitado em Junho de 2010).
<http://www.htc.com/www/product/dream/specification.html>
 28. Meier R. (2009), Professional Android Application Development. Wiley Publishing, Inc. pp. 323-333.
 29. Datasheet do acelerómetro de 3 eixos Bosch Sensortec BMA150. 2008, Bosch.
 30. Datasheet da bússola digital de 3 eixos AK8973. 2007, A. Kasei.
 31. Eclipse (visitado em Junho de 2010). <http://www.eclipse.org/>
 32. Microsoft. Microsoft Visual Studio (visitado em Junho de 2010).
<http://msdn.microsoft.com/en-us/vstudio/default.aspx>.
 33. Schroeder W. (2006), The VTK User's Guide, 5th edition. Kitware, Inc.
 34. Stevens W., Fenner B., Rudoff A. (2003), UNIX Network Programming Volume 1, Third Edition: The Sockets Networking API. Addison-Wesley Professional. pp. 28-52.
 35. Strickland J., What is a gimbal -- and what does it have to do with NASA? (visitado em Junho de 2010) <http://science.howstuffworks.com/gimbal.htm/printable>
 36. Wikipedia, Gimbal Lock (visitado em Junho de 2010).
http://en.wikipedia.org/wiki/Gimbal_lock.
 37. Nielsen J. (1994), Usability Engineering. Morgan Kaufmann. pp. 207-213.
 38. Dix A., Finlay J., Abowd G., Beale R. (2004), Human-Computer Interaction, 3rd edition. Prentice Hall.

39. ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs); Part 11 - Guidelines for specifying and measuring usability, in International Standards Organisation. Genève.
40. Limin F., Salvendy G., Turley L. (2002), Effectiveness of user testing and heuristic evaluation as a function of performance classification. Behaviour and Information Technology. Vol. 21.
41. Nielsen J., Ten Usability Heuristics (visitado em Junho de 2010). <http://www.useit.com/papers/heuristic>.
42. Rienman J., Frankze M., Redmiles D. (1995), Usability Evaluation with the Cognitive Walkthrough, in Conference companion on Human factors in computing systems. (Denver, USA: ACM), pp. 387-388.